

**UNIVERSIDAD
AUTONOMA DE MADRID
ESCUELA POLITECNICA SUPERIOR**



TRABAJO DE FIN DE GRADO

**DESARROLLO DE UNA APLICACIÓN
DE ENTRENAMIENTO Y EVALUACIÓN DE LA ATENCIÓN
EN NIÑOS Y ADOLESCENTES CON TDAH
A TRAVES DE DISPOSITIVOS MULTICONTACTO**

Lucia Gómez López
(lucia.gomezl@estudiante.uam.es)

SEPT 2013

**DESARROLLO DE UNA APLICACIÓN
DE ENTRENAMIENTO Y EVALUACIÓN DE LA ATENCIÓN
EN NIÑOS Y ADOLESCENTES CON TDAH
A TRAVES DE DISPOSITIVOS MULTICONTACTO**

AUTORA: Lucia Gómez López
TUTORA: Rosa M^a Carro Salas

Escuela Politécnica Superior
Universidad Autónoma de Madrid
SEPT 2013

Resumen — A diario convivimos con las nuevas tecnologías en multitud de ámbitos de nuestra vida; utilizamos dispositivos móviles, tabletas, ordenadores de sobremesa o portátiles, etc. A veces hasta nos cuesta imaginarnos cómo sería nuestra vida sin los avances tecnológicos que conocemos actualmente. Además de facilitarnos la vida y hacer más cómodas muchas de las tareas de la vida cotidiana, las nuevas tecnologías poco a poco se abren camino en otros campos no tan directamente relacionados con la informática y las telecomunicaciones, como la medicina. Esto nos plantea continuos retos relacionados con el modo de integrar los nuevos avances en forma de trabajar ya existente y nos invita a buscar posibles necesidades que se puedan satisfacer mediante el desarrollo e implantación de nuevas aplicaciones específicas. Añadido al reto técnico nos encontramos con la dificultad de integrar nuevos dispositivos o aplicaciones en procesos que hasta el momento se han llevado a cabo de una forma determinada, y hay que tener en cuenta que dependiendo del objetivo, este cambio afectará a colectivos de personas de distintos tipos, edades y formación. Es fundamental la colaboración de especialistas de los campos involucrados en proyectos de este tipo para que constituyan un éxito, tanto a nivel técnico como en cuanto al rigor científico, que es fundamental para maximizar la utilidad real del proyecto y su aplicación en el campo en cuestión. En concreto este proyecto se centra en el desarrollo de una aplicación para utilizar durante la terapia como método de entrenamiento y evaluación con niños y adolescentes con TDAH (Transtorno por Déficit de Atención e Hiperactividad) sobre una mesa multicontacto mediante la que, a través del juego, se ayudará al paciente a mejorar la capacidad de concentración y control de impulsos. Se ha contado con la ayuda de un psicólogo especialista en la materia¹ para dotar al proyecto del rigor necesario. Su experiencia ha permitido plasmar los fundamentos teóricos en la aplicación práctica, de modo que finalmente se ha conseguido desarrollar una aplicación útil para una terapia real.

Palabras Clave — Aplicaciones para dispositivos multicontacto, atención a la diversidad, usuarios con TDAH, mejora de la atención, control de impulsos, terapia a través del juego.

¹Ángel Felipe Nieto, Psicólogo licenciado en la Universidad Española de Educación a Distancia (UNED). 4 años de experiencia en terapia con pacientes con TDAH.

Abstract — New technologies are a regular occurrence in our daily lives. We sometimes can't imagine our life without the technological advances that take for granted. In addition to make our lives easier and many tasks more comfortable, new technologies are slowly opening roads in other fields not directly related with computing and telecommunications, like medicine. This exposes us to continuous challenges for integrating the new advancements with existing ways of working and for finding needs that we can satisfy through the development and implementation of new applications. Added to technical challenge we find troubles integrating new devices or applications in processes that have been carried out in a particular way so far, and we need to take into account that depending of the type of project adding this technological layer might affect groups of people of all type, age and formation. When this happens the cooperation with specialists in the matters involved is critical to be successful, both in technical level and scientific rigor and becomes essential for maximizing the project's real utility and it's applications. Specifically this project focuses on an application development for use in therapy with children and teenagers with ADHD (Attention-Deficit Hyperactivity Disorder) on a multitouch device. Through games, we help patients improve their ability on concentration and impulse control. We rely on an expert psychologist specialized in this condition² to give the project a strong scientific foundation so that it can become a useful tool in real therapy.

Index Terms — Multitouch devices applications, ADHD patients, psychology, attention improvement, impulse control, therapy through games.

²Ángel Felipe Nieto, Licensed Psychologist in the Spanish University for Distance Education (UNED). 4 years of experience in patients with ADHD therapy.

Agradecimientos

Me resulta imposible al echar la vista atrás no acordarme de muchas personas que a lo largo de mi vida han estado a mi lado y han contribuido día a día a que haya podido llegar hasta aquí y culminar una etapa tan importante. Creo que no sería justo no dedicarles unas líneas en este trabajo. Por ello, para mí esta página es una de las más importantes, si no la que más.

En primer lugar quiero agradecerle a Rosa, mi tutora, toda su dedicación e ilusión por este trabajo. Gracias porque desde el principio me has transmitido apoyo, optimismo y ganas que han sido fundamentales para terminarlo con éxito.

A Ángel, sin cuya ayuda no hubiera sido posible este trabajo. Gracias por su paciencia, su humor y todas las aportaciones que ha realizado.

A la empresa Tecnológica y a todos los compañeros que me habéis echado una mano o, simplemente os habéis interesado por el proyecto.

A todas las personas que de una forma u otra habéis contribuido al proyecto, gracias por vuestra ayuda, especialmente a Isa por su colaboración en las pruebas.

A mi familia, en especial a mis padres y mi hermano, por haber estado siempre ahí, en días buenos y malos para ofrecer vuestro apoyo y cariño incondicional, vuestra sinceridad y vuestra experiencia.

A todos los que a lo largo de mi vida me habéis enseñado algo; profesores del colegio, universidad, *Scouters*, entrenadores, etc. Porque de cada uno de vosotros he podido aprender una cosa, y la suma de todas ellas me han ayudado a ser la persona que soy hoy. Con un recuerdo especial para Juanra; aunque ya no estés con nosotros, tus enseñanzas más allá de las matemáticas siguen más que presentes en nuestras vidas.

A todos mis compañeros de carrera por haber hecho de esta etapa una de las más importantes y a la vez más divertidas de mi vida.

A todos mis amigos, por estar ahí tantos y tantos años. Por vuestro apoyo, comprensión y por todos los momentos importantes en los que habéis estado conmigo.

A Diego, por todos y cada uno de los días que han sido mejores gracias a que has estado a mi lado con tu sonrisa, tus bromas y tu forma de ser.

Finalmente, me gustaría dedicar este trabajo a una personita que por desgracia se ha ido demasiado pronto. Para ti, Laurita, que sigas sonriendo como hiciste toda tu vida y allá donde estés, no dejes nunca de jugar.

Índice general

1	Introducción	1
1.1	Objetivos y retos	2
1.2	Introducción al TDAH	2
1.3	Fases de realización	3
1.4	Estructura del documento	4
2	Estado del Arte	5
2.1	Introducción	5
2.2	Trabajo previo de investigación	5
2.3	Descripción de la aplicación base	7
2.3.1	Aspectos mejorables	9
3	Análisis de requisitos	11
3.1	Introducción	11
3.2	Requisitos funcionales	11
3.2.1	Gestión de usuarios	11
3.2.2	Gestión de sesiones	13
3.2.3	Gestión de resultados	14
3.2.4	Gestión del juego	15
3.2.5	Descripción del juego	16
3.3	Requisitos no funcionales	19
4	Diseño de la solución	21
4.1	Arquitectura de la aplicación	21
4.2	Descripción de los módulos	24
4.2.1	Lógica general de la aplicación	25
4.2.2	Lógica de los módulos	25
4.3	Descripción de la base de datos	29
4.4	Interfaz gráfica	31
4.5	Conclusiones	33
5	Implementación	35
5.1	Estructura del código	35
5.2	Módulos de comunicación	37
5.3	Estructura de los ficheros XML	37
5.4	Interfaz gráfica mediante MT4J	41
5.5	Conclusiones	42

6	Implantación y pruebas	45
6.1	Implantación	45
6.2	Plan de pruebas y resultados	46
6.2.1	Pruebas unitarias	47
6.2.2	Pruebas de integración	48
6.2.3	Pruebas del sistema	48
6.2.4	Pruebas con usuarios	50
7	Conclusiones y trabajo futuro	53
A	Prototipo de interfaz mejorada	57
B	Tablas de atributos de las clases	59

Índice de cuadros

3.1	Niveles definidos para el JCS	17
B.1	Atributos de la clase Sistema	59
B.2	Atributos de la clase Usuario	60
B.3	Atributos de la clase Historial	60
B.4	Atributos de la clase InfoResultado	60
B.5	Atributos de la clase RegistroTiempo	60
B.6	Atributos de la clase Sesion	61
B.7	Atributos de la clase Juego	61
B.8	Atributos de la clase Nivel	61
B.9	Atributos de la clase MotorJuego	62

Índice de figuras

2.1	Imagen del juego “ <i>Brain training</i> ”	6
2.2	Portada de la plataforma NeuronUP	7
2.3	Captura del arranque de la aplicación CAPTAIN	8
2.4	Captura durante el desarrollo del juego base	8
2.5	Captura de la presentación de resultados del juego al usuario	9
3.1	Diagrama explicativo de los criterios de cambio de nivel	18
4.1	Arquitectura del sistema	22
4.2	Diagrama de clases sin detallar de la aplicación	25
4.3	Diagrama de clases del módulo sistema	26
4.4	Diagrama de clases del módulo de gestión de usuarios	26
4.5	Diagrama de clases del módulo de gestión de resultados	27
4.6	Diagrama de clases del módulo de gestión del juego	28
4.7	Esquema de la pantalla panel de usuarios	32
4.8	Esquema de la pantalla del juego	33
5.1	Datos básicos de un usuario en <i>listadoUsuarios.xml</i>	38
5.2	Información completa de un usuario	38
5.3	Información de sesión	39
5.4	Campos de información de resultados del juego	39
5.5	Almacenamiento de resultados del ejercicio	40
5.6	Listado de juegos que maneja la aplicación	40
5.7	Fichero de configuración del JCS	40
5.8	Captura de pantalla del panel de usuarios	41
5.9	Captura de pantalla del JCS	42
6.1	Comparativa de capturas del panel de usuarios	46
6.2	Ejemplo de gráfico circular a partir de resultados	47
6.3	Pruebas del juego en línea de comandos	49
A.1	Prototipo de la interfaz mejorada	57
A.2	Prototipo de la interfaz mejorada	58
A.3	Prototipo de la interfaz mejorada	58
A.4	Nueva combinación de colores y tipografía	58

Glosario

ADHD En Inglés, siglas equivalentes a TDAH.

API Conjunto de métodos y procedimientos que ofrece una biblioteca. Se utilizan a modo de documentación de las mismas.

CAPTAIN Aplicación comercializada en 1985 que se utiliza en la actualidad en terapia real con pacientes con, entre otros, TDAH.

Eclipse Entorno de programación que se ha utilizado para desarrollar el proyecto.

Java Lenguaje de programación orientado a objetos que se ha utilizado para el desarrollo de la aplicación.

Javabeans Modelo de componentes para el lenguaje java que permite utilizar los objetos como parte de la base de datos.

JCS Juego de Coincidencia Simbólica. Juego desarrollado en la aplicación TDAApp.

JDOM Biblioteca utilizada en el desarrollo de la aplicación para la lectura y escritura de archivos XML.

JFreeChart Biblioteca que proporciona los métodos necesarios para la construcción de gráficas a partir de conjuntos de datos.

Kepler Versión del entorno de programación Eclipse que se ha utilizado durante el desarrollo.

MT4J Librería utilizada en el proyecto para la construcción de la interfaz gráfica.

SUR40 Mesa multicontacto fabricada por Samsung y Microsoft que se ha utilizado a lo largo del desarrollo del proyecto.

TDAH Transtorno por Déficit de Atención e Hiperactividad.

TDAApp Nombre de la aplicación desarrollada en este proyecto.

XML Lenguaje de marcas utilizado en la aplicación para el almacenamiento de datos.

1

Introducción

En los últimos años la tecnología ha avanzado a un ritmo vertiginoso en multitud de campos. Con algunos de ellos convivimos a diario, por ejemplo los dispositivos móviles de última generación: *smartphones* con los que podemos llevar a cabo un gran número de acciones; desde una llamada telefónica hasta hacer la compra con una simple aplicación y conexión a Internet. Continuamente se lanzan nuevas aplicaciones con las que podemos hacer aún más cosas desde la palma de nuestra mano contribuyendo a hacer más fáciles y cómodas muchas de las tareas de la vida cotidiana y abriendo la puerta al desarrollo de nuevas ideas que den respuesta a las necesidades que demanda la sociedad.

De entrada podría parecer que todo son ventajas alrededor de este avance tan rápido. No obstante, si reflexionamos durante un momento podemos encontrar inconvenientes que también es importante tener en cuenta. Uno de ellos es que no todo el mundo asimila igual los cambios, ya sea debido a la edad, la formación u otros factores, a muchas personas les cuesta mucho adaptarse a las nuevas formas de hacer las cosas, tan distintas de como las han hecho hasta ahora. Sin embargo, poco a poco se está consiguiendo llegar al mayor número posible de personas y hacerlas partícipes de este “*Boom* de las nuevas tecnologías”.

Sin embargo, lo que es indudable es que las ventajas superan ampliamente a los inconvenientes; gracias a las nuevas tecnologías se ha conseguido por ejemplo, contribuir a grandes avances en la ciencia y en la medicina, mejorar sustancialmente la calidad de vida de personas con necesidades especiales mediante aplicaciones específicas e incluso la construcción de dispositivos que permitan a personas con algunas discapacidades físicas realizar acciones que de otra forma les hubieran resultado imposible.

Precisamente esto es lo que nos ha llamado la atención a la hora de plantear este proyecto: cómo podemos utilizar las nuevas tecnologías para ayudar a grupos específicos de personas. En concreto nos hemos centrado en los niños y adolescentes con TDAH (más adelante explicaremos en qué consiste) y como podemos hacer más atractiva y efectiva la terapia. La respuesta la encontramos en un dispositivo cuyo uso aún no está muy extendido: la mesa multicontacto y el desarrollo de una aplicación específica para utilizar en la terapia llevada a cabo con estos pacientes.

1.1 Objetivos y retos

La primera idea al pensar en posibles objetivos para este Trabajo de Fin de Grado fue la de diseñar y desarrollar una aplicación que ayudase a colectivos de personas con algún tipo de trastorno o discapacidad. Finalmente nos decantamos por el Trastorno por Déficit de Atención e Hiperactividad, ya que es un tipo de trastorno que afecta especialmente a niños y adolescentes siendo especialmente importante hacer atractiva la terapia para ellos.

Al contactar con un psicólogo especialista en la materia, vimos que ya hay algunas aplicaciones para trabajar con personas que padecen este trastorno, aunque como veremos en el capítulo 2 las que se han venido utilizando los últimos años comienzan a quedarse anticuadas para los nuevos sistemas operativos y dispositivos y faltan desarrollos que aprovechen el potencial de estos últimos.

El objetivo principal de este Trabajo de Fin de Grado es crear una aplicación ejecutable sobre dispositivos multicontacto orientada al entrenamiento de niños y adolescentes con TDAH que pueda ser utilizada como parte de la terapia.

Se basará en “CAPTAIN”, una aplicación existente que se detalla en el apartado 2.3, detallándose en el apartado 2.3.1 los aspectos mejorables de dicha aplicación que nos condujeron a plantear este proyecto.

Dado que se ha decidido desarrollar la aplicación sobre una mesa multicontacto, uno de los retos para la autora del proyecto es la investigación del desarrollo en este dispositivo: arquitectura de las aplicaciones, librerías, etc. Supone un reto a nivel técnico ya que no he desarrollado anteriormente nada parecido. También supone un desafío en cuanto al hecho de que, al ser una aplicación para un colectivo específico hay que diseñar algunas partes teniendo en cuenta aspectos ajenos a lo técnico, como la interfaz gráfica. Más adelante detallaremos cómo se han abordado estos aspectos.

Finalmente, otro de los retos a destacar es el de llevar a cabo un proyecto desde cero, planteando todas sus fases desde el análisis de requisitos hasta la implementación y pruebas, planificando tiempo y recursos y poniendo en práctica la mayoría de los conocimientos y capacidades adquiridos a lo largo de la carrera.

1.2 Introducción al TDAH

Dado que se va a desarrollar una aplicación orientada a usuarios con este trastorno, es importante dar una visión general de en qué consiste y que supone para entender mejor lo que queremos construir.

El Trastorno por Déficit de Atención e Hiperactividad es uno de los trastornos de conducta más frecuente en la infancia y la adolescencia, con una prevalencia estimada entre el 3 % y el 5 % de la población [1], que consiste en un alto nivel de desatención (dificultad particularmente alta para mantener la atención en un estímulo o actividad), hiperactividad (hipercinesia, impo-

sibilidad de estar quieto o sentado durante periodos breves, habla continua...) e impulsividad (precipitación, interrupciones, falta de control) [2].

Saliendo del lenguaje más académico, las personas que presentan este síndrome son muy inquietas, se aburren con facilidad, les cuesta atender en clase o en el trabajo y evitan llevar a cabo tareas que les requieran mucha dedicación y paciencia, porque necesitan desahogar de alguna forma toda la activación que tienen.

Tampoco son capaces de mantener el control en determinadas situaciones y se lanzan a acciones o tomas de decisiones sin tener en cuenta las consecuencias o parte de la información que tienen.

Clínicamente se suele llevar a cabo un tratamiento con dos o tres pautas:

- **Neuroquímica:** A pesar de la nomenclatura única, las causas neurológicas varían de unos casos a otros y esto implica que se pueda tratar con neuroestimulantes (para la activación de las zonas del córtex que se encargan del control de impulsos), o neurodepresores para la reducción de activación.
- **Conductual:** Se trata de dar pautas y enseñar a la persona a manejar su situación, en muchos casos con actividades organizadas para que “entrene” su propia capacidad de control de la atención y los impulsos, yendo de actividades más sencillas a más complicadas y consiguiendo un aprendizaje a nivel neurológico que reduzca el trastorno.
- **De entorno:** Generalmente organizando el entorno en el que vive y enseñando a padres de pacientes menores de edad a manejar las situaciones y a educar de forma adecuada.

Dentro de la segunda, la conductual, se enmarcan todas las aplicaciones y ejercicios destinados al tratamiento del TDAH en terapia psicológica, y por lo tanto, en donde podemos clasificar la aplicación que se desarrolla en este Trabajo de Fin de Grado.

1.3 Fases de realización

En este apartado vamos a dar una visión general de cuales son las fases de realización del proyecto, las cuales se han llevado a la práctica de la forma más rigurosa posible para intentar garantizar al máximo el éxito del proyecto.

Las fases de realización en este proyecto son las correspondientes a la mayoría de proyectos de desarrollo de software:

1. **Análisis:** En esta fase se va a plasmar en forma de requisitos la idea que tenemos para la aplicación. Con esto conseguiremos saber qué es exactamente lo que queremos construir y acotaremos los límites de la aplicación.
2. **Diseño:** Una vez concluida la fase de análisis procederemos a diseñar la arquitectura del producto final, dividirlo en módulos agrupados por funcionalidad común, y a establecer como se comunicarán entre ellos. También se decidirá en esta fase cual será el lenguaje de programación que se utilizará, cómo se almacenarán los datos y otros aspectos de diseño que serán convenientemente justificados.

3. **Implementación:** Programación de la aplicación en el lenguaje decidido en la fase anterior. En esta fase se realizarán también las pruebas unitarias de cada uno de los módulos y las de integración de unos módulos con otros.
4. **Pruebas e implantación:** En esta fase se llevarán a cabo las pruebas generales de la aplicación y la implantación de la misma en el dispositivo multicontacto *Samsung SUR40*. Se realizarán asimismo pruebas con pacientes de TDAH para comprobar la efectividad y utilidad del software desarrollado.
5. **Documentación:** Es fundamental en el desarrollo de cualquier aplicación dedicar tiempo y esfuerzo a la documentación del mismo, tanto a nivel de código como para plasmar por escrito el resultado de cada una de las fases ya comentadas.
6. **Mantenimiento:** En este proyecto esta es la fase que puede tener menos sentido, pero la orientamos a un posible trabajo futuro para ampliar y mejorar la aplicación.

1.4 Estructura del documento

Esta memoria recoge todas las fases descritas en el apartado anterior.

En el siguiente capítulo se ha realizado una pequeña investigación del Estado del Arte para saber qué es lo que podemos encontrar actualmente relacionado con terapias basadas en el ordenador para el TDAH y de que forma se puede mejorar.

Posteriormente abordaremos cada una de las fases de desarrollo que se han comentado en el apartado anterior.

En el capítulo 3 se recoge todo el análisis de requisitos de la aplicación. Su comprensión es fundamental para entender en que consiste el producto final.

En el capítulo 4 se explica todo el diseño de la aplicación incluyendo la arquitectura, casos de uso, descripción de los módulos, bases de datos y decisiones que se han tomado para la siguiente fase.

El capítulo 5 documenta la implementación del diseño realizado en el lenguaje de programación elegido y las pruebas realizadas en esta fase.

En el capítulo 6 se detallará que pruebas se han realizado a la aplicación en conjunto y como se ha implantado la aplicación en la mesa multicontacto.

Finalmente, en el capítulo 7 se presentarán las conclusiones una vez realizado el proyecto y se plantearán las posibles mejoras que se puedan aportar al proyecto una vez finalizado este Trabajo de Fin de Grado.

2

Estado del Arte

2.1 Introducción

En este capítulo se van a presentar los resultados de una pequeña investigación que hemos llevado a cabo en el campo de la informática aplicada a la psicología, en especial aplicaciones específicas para TDAH. Es importante antes de hacer una nueva aplicación ver que es lo que hay ya desarrollado puesto que así no solo se puede ahorrar mucho trabajo al no trabajar en cosas que ya existan, sino que mucho más importante, se pueden encontrar carencias y posibilidades de mejora, detectando así como se puede contribuir en esta línea.

Además, dado que la aplicación desarrollada en este trabajo toma la base de otra ya existente aunque muy antigua, se dedicará un apartado de este capítulo a hacer una breve descripción de dicha aplicación, mostrando sus posibilidades actualmente, sus limitaciones y sus posibilidades de mejora, que serán abordadas.

2.2 Trabajo previo de investigación

En este apartado se va a dar una visión general del software específico desarrollado para el campo de la psicología. Como ya se ha comentado, es importante realizar este trabajo previo al desarrollo de cualquier aplicación ya que la originalidad y la innovación son clave para maximizar el éxito de ésta.

Existen multitud de aplicaciones orientadas al campo de la psicología, como se puede ver en [3]. En esta página web se puede encontrar uno de los más amplios repositorios de programas específicos, algunos de ellos gratuitos. Se puede observar que los programas gratuitos están orientados a la gestión de clínicas o gabinetes de psicología, mientras que los de pago tratan específicamente algún tipo de problema, como puede ser el estrés.

Sin embargo, no todas las aplicaciones son únicamente utilizadas en un entorno profesional, como es el caso del famoso juego “*Brain training*”.



Figura 2.1: Imagen del juego “Brain training”

Este juego, comercializado en el año 2006 por la empresa Nintendo, consta de pequeñas pruebas en las que se van planteando al jugador pequeños retos que ayudan a trabajar y potenciar diferentes capacidades: comprensión verbal, razonamiento matemático, visión espacial, etc. Se plantea en niveles en los que a medida que el usuario avanza aumenta la dificultad.

En Internet se pueden encontrar multitud de plataformas con ejercicios y juegos similares a los del “Brain training”, como es el caso de “Brain arena” [4], donde se puede encontrar un recopilatorio de más de 1000 juegos para entrenar la mente completamente gratuito, que permite además al usuario crear una cuenta personal que puede incluso conectar con su Facebook.

Sin embargo, alrededor de este tipo de aplicaciones han surgido opiniones de todo tipo. Mientras que multitud de expertos afirman que su uso regular ayuda a mejorar las capacidades y retrasar el deterioro de las funciones del cerebro, en el otro lado encontramos estudios que niegan la base científica y afirman que no se ha demostrado que estos juegos tengan un beneficio mayor que otras actividades que estimulen a la persona o paciente de una forma similar.

Donde se puede encontrar menor división de opiniones es en otro tipo de aplicaciones, desarrolladas específicamente para grupos de personas con algún tipo de discapacidad o enfermedad degenerativa como el Alzheimer, enseñándoles o ayudándoles a llevar a cabo actividades de la vida cotidiana. Mediante el uso de estas aplicaciones se pretende mejorar la calidad de vida de estas personas o retrasar los efectos de algunas de las enfermedades que afectan al cerebro. Al estar desarrolladas específicamente para estos colectivos se tiene mucho más en cuenta las características de los usuarios finales y sus necesidades, por lo que podríamos hablar de aplicaciones hechas “a medida”.

Un ejemplo de este tipo de aplicación es “NeuronUP- [5]. Esta plataforma, lanzada en 2012 está construida por un equipo formado por informáticos, neuropsicólogos y diseñadores, y a las dos semanas de su lanzamiento fue adquirida por la Federación Española de Daño Cerebral (Fedace). Consta de más de 6000 ejercicios, y cada uno de ellos tiene un propósito determinado, diferenciando entre niños y adultos.



Figura 2.2: Portada de la plataforma NeuronUP

Ademas comentar que esta plataforma es accesible *online* desde cualquier navegador (en la figura 2.2 se puede ver la portada *online*), cuenta con aplicaciones para Android, iOS y algunos de sus ejercicios y juegos han sido adaptados para la Kinect de Microsoft, lo que supone un gran avance en cuanto a la interacción de la persona con el programa, mucho más sencillo y seguramente más efectivo.

De todas estas aplicaciones que se han comentado a lo largo de este apartado no se ha encontrado ninguna específica para terapia con TDAH salvo la aplicación base que se detallará en el siguiente apartado.

2.3 Descripción de la aplicación base

Como ya se ha comentado anteriormente, la idea para este trabajo toma como base parte de un programa ya existente que, a pesar de ser bastante antiguo, se sigue utilizando en terapia con pacientes con TDAH. En este apartado se va a dar una visión general de esta aplicación y se explica el punto de partida de nuestro proyecto, dejando claro cuales son las partes que se basan en dicha aplicación y qué partes son nuevas.

Dicha aplicación, llamada CAPTAIN data del año 1985 y está programada en lenguaje ensamblador. Debido a su antigüedad ha sido necesario ejecutarla en una máquina virtual con Windows XP de 32 bits para poder manejarla y entender su funcionamiento. En la figura 2.3 se muestra una captura de pantalla del arranque.

CAPTAIN cuenta con más de 50 ejercicios, en los que a través del juego se potencian y trabajan capacidades. Los menús están estructurados por la habilidad con la que queremos trabajar.



Figura 2.3: Captura del arranque de la aplicación CAPTAIN

En concreto nos proponemos tomar como idea el juego “Pattern display match” que se encuentra en la categoría “Conceptual skills” y es utilizado en terapia real con niños y adolescentes con TDAH. El funcionamiento del juego es simple: se presentan dos imágenes al usuario, que deberá decir si éstas coinciden o no de acuerdo con el criterio actual (que puede ser color, forma o ambos). A continuación se puede ver una captura de pantalla del juego en la figura 2.4.

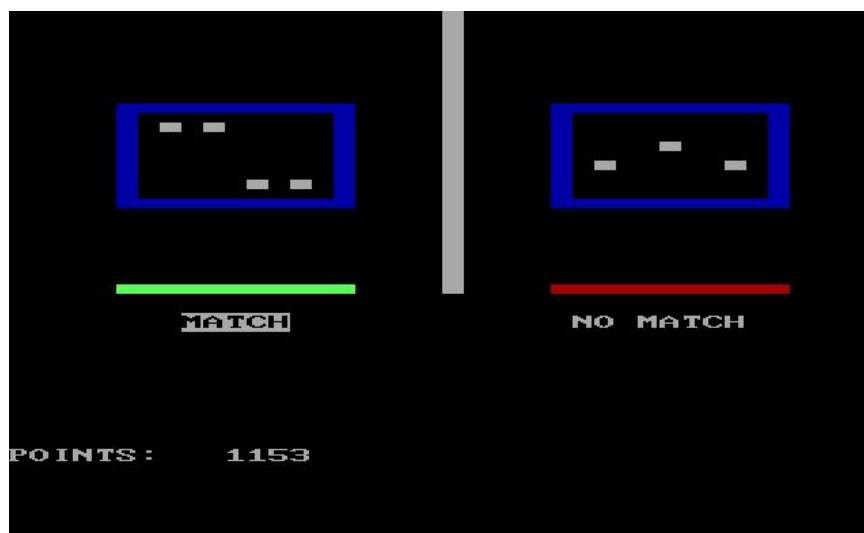


Figura 2.4: Captura durante el desarrollo del juego base

Finalmente en la figura 2.5 se muestra la presentación de resultados al usuario al finalizar el juego, otro de los aspectos que nos proponemos mejorar y ampliar.

Cada vez que se juega, el terapeuta configura ciertos aspectos que afectan al juego, como la duración, la dificultad y variables como:

- Ocultación automática de las imágenes transcurrido un tiempo determinado.

- Aviso al paciente del cambio de criterio cuando este se produce.
- Mostrar el criterio actual en la pantalla.
- Configuración de un tiempo máximo de respuesta.

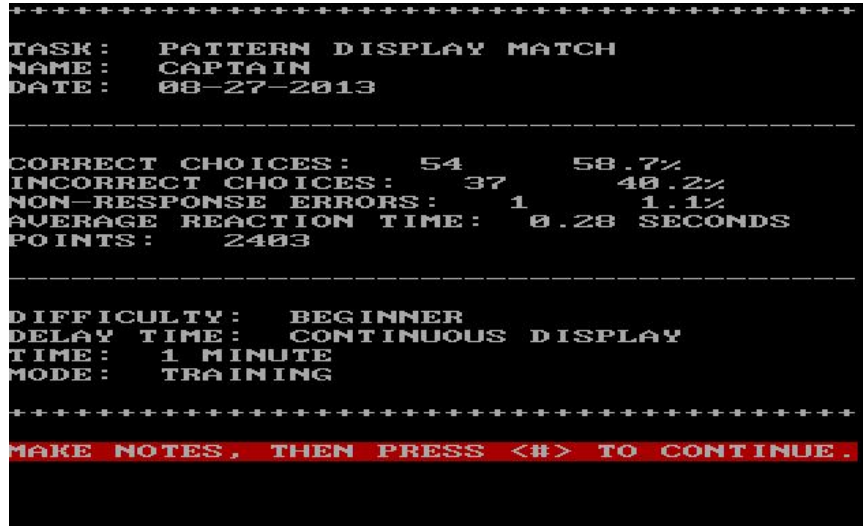


Figura 2.5: Captura de la presentación de resultados del juego al usuario

En el juego original la combinación de estas y otras variables son configurados durante el desarrollo de los juegos por el psicólogo que lleva a cabo la terapia, que debe decidir en el momento ayudado por el historial del usuario. En este proyecto se pretende plasmar en la aplicación el conocimiento del experto de modo que no sea necesaria una configuración manual antes de la ejecución del juego: mediante el establecimiento de niveles, cada uno de los cuales combina de una forma u otra todos los parámetros. En el capítulo 4 se detallarán todos estos aspectos.

2.3.1 Aspectos mejorables

En este apartado se van a comentar cuales son algunos de los aspectos que se pretenden mejorar al construir una nueva aplicación que toma sus bases:

- La interacción mediante dispositivo multicontacto permite eliminar el posible tiempo de retraso en el que el paciente resuelve cada ejercicio, dado que necesita llevar el ratón a la opción correcta.
- Eliminación de las barreras a los pacientes con problemas de motricidad al prescindir del ratón.
- Automatizar la recogida de datos y completar las mediciones y estadísticas según la experiencia obtenida en terapia respecto a la anterior aplicación.
- Añadir un sistema de gestión de usuarios y sesiones que permita al terapeuta consultar la información de los pacientes en cualquier momento.
- Crear una nueva interfaz gráfica que sea más agradable a la vista del paciente.

- Ofrecer la posibilidad de presentar en una misma superficie la interfaz para el paciente y la interfaz con la monitorización en vivo para el terapeuta.

En el siguiente capítulo se realizará un análisis de requisitos, como parte fundamental para terminar de comprender qué es lo que se va a construir.

3

Análisis de requisitos

3.1 Introducción

En este capítulo se presentan los requisitos de la futura aplicación. En el siguiente capítulo se tomará este conjunto de requisitos como base para el diseño de todos los aspectos de la misma, desde la estructura de la base de datos hasta la colocación de los elementos en cada una de las pantallas que compondrán la interfaz gráfica. Es por eso que la fase de análisis tiene una gran importancia dentro del proceso de desarrollo de software; se sientan las bases del proyecto a partir de las cuales construimos todo lo demás, de la misma forma que los cimientos de una casa: si se hacen mal, la casa se derrumba.

La comprensión de este capítulo es fundamental para entender la aplicación que se desarrolla en este proyecto. A continuación se plantea la funcionalidad y características del mismo en forma de análisis de requisitos funcionales y no funcionales.

3.2 Requisitos funcionales

En este apartado se presentarán los requisitos funcionales del producto, es decir, aquellos que describen directamente la funcionalidad del programa: su comportamiento.

Para una mejor estructuración del catálogo de requisitos, se van a presentar en subapartados, agrupándolos por la parte del sistema que definen: gestión de usuarios, de sesiones, de resultados y del juego. Finalmente se realizará una descripción del juego que se va a implementar como parte de la aplicación.

3.2.1 Gestion de usuarios

En este apartado se van a exponer los requisitos relacionados con la gestión de usuarios y los datos de éstos. Esta es una de las mejoras sustanciales con respecto al programa original, que no cuenta con ningún tipo de almacenamiento de datos relativos a usuarios ni da soporte a la

gestión de los mismos.

La idea es que el terapeuta pueda crear un usuario para cada uno de los pacientes con los que lleve a cabo la terapia en la que se utilice la aplicación. Se almacenarán los resultados de cada una de las sesiones de terapia automáticamente, lo que permitirá ir generando también automáticamente un historial de resultados que le sirvan para analizar la evolución del paciente a lo largo de las sesiones.

Los datos que se almacenarán de cada usuario son: nombre, apellidos, fecha de nacimiento, foto de perfil, diagnóstico, fecha de inicio de la terapia y observaciones del terapeuta. Además se ligará al perfil del usuario toda la información de cada una de las sesiones llevadas a cabo y de cada uno de sus juegos.

Los requisitos funcionales que se definen en relación con los usuarios son:

- **RF1.1:** Crear nuevo usuario en el sistema.
 1. Dentro del panel de gestión de usuarios se seleccionará la opción "Nuevo usuario".
 2. El sistema solicitará los datos. Los datos obligatorios para dar de alta un nuevo usuario son: nombre, apellidos y fecha de nacimiento. El resto de datos son opcionales y podrán aportarse en este momento o más adelante.
 3. Una vez finalizada la introducción de datos, el sistema almacenará en la base de datos el nuevo usuario.
- **RF1.2:** Consultar los datos del usuario.
 1. Dentro del panel de gestión de usuarios se seleccionará un usuario y la opción "Consultar datos".
 2. Se abrirá la pantalla de visualización de datos en la que se pueden ver todos los datos guardados del usuario.
- **RF1.3:** Modificar datos de un usuario existente en el sistema.
 1. Dentro del panel de gestión de usuarios se seleccionará un usuario y la opción "Consultar datos".
 2. En la pantalla de visualización de datos se seleccionará la opción "Modificar datos".
 3. Se modificarán los datos que se desee en los campos editables y se seleccionará "Guardar cambios".
 4. El sistema comprobará que sigue teniendo todos los datos obligatorios y guardará en la base de datos los nuevos datos del usuario.
- **RF1.4:** Eliminar un usuario del sistema.
 1. Dentro del panel de gestión de usuarios se seleccionará un usuario y la opción "Consultar datos".
 2. En la pantalla de visualización de datos se seleccionará la opción "Eliminar usuario".

3. Al ser una operación irreversible, el sistema pedirá confirmación explicando al usuario que no se pueden recuperar los datos borrados.
 4. En caso de respuesta afirmativa se eliminarán todos los datos del usuario, así como los datos asociados de sesiones y resultados.
- **RF1.5:** Consultar otros datos asociados al usuario.
 1. Dentro del panel de gestión de usuarios se seleccionará un usuario y la opción “Consultar datos”.
 2. En la pantalla de visualización de datos se seleccionará la opción “Consultar sesiones”.
 3. Se visualizarán los datos de sesiones y resultados de las mismas. Se darán más detalles de esta funcionalidad en el apartado correspondiente a “Visualización de resultados”.

3.2.2 Gestión de sesiones

Continuando con las mejoras sobre el programa original, se añade un sistema de sesiones, que almacenará los resultados de los usuarios cada vez que jueguen. Antes de continuar exponiendo los requisitos es fundamental definir los términos que se van a utilizar en este apartado y en los siguientes:

- **Sesión:** Corresponde a una sesión real en terapia, en la que el terapeuta puede realizar uno o varios juegos con el paciente. Los resultados obtenidos en cada uno de ellos se almacenarán automáticamente en el sistema.
- **Juego:** Cada uno de los juegos que se llevan a cabo en la sesión. En una sesión se puede llevar a cabo un juego una o varias veces.
- **Ejercicio:** Cada una de las partes de las que consta un juego. En cada uno de los juegos se detallará en que consiste el ejercicio correspondiente.

Los datos que se guardan de la sesión son los siguientes: fecha y hora de inicio, duración, comentarios del terapeuta y toda la información de los juegos llevados a cabo en dicha sesión. Además la sesión debe quedar asociada con el usuario. En el capítulo 4 se detallará como se implementan estas referencias.

Los requisitos funcionales correspondientes a la parte de gestión de sesiones son:

- **RF2.1:** Crear una nueva sesión asociada a un usuario.
 1. En el panel de gestión de usuarios, se seleccionará la opción “Nueva sesión”.
 2. El sistema creará automáticamente la nueva sesión rellenando la fecha y la hora de inicio de la misma. Asimismo, asociará la sesión al usuario. En este momento no se guardará nada en la base de datos, hasta que no se guarden los resultados de un juego.
- **RF2.2:** Finalizar una sesión.
 1. En el panel de gestión de la sesión se seleccionará la opción “Finalizar sesión”.

2. Puesto que es una operación irreversible, el sistema informará de ello al usuario y pedirá su confirmación antes de seguir adelante.
3. En caso afirmativo se completan el resto de datos de la sesión: la duración (lo calcula el sistema a partir de la hora de inicio) y ofrece al terapeuta la posibilidad de guardar información que añada en forma de comentarios.
4. Se guarda la sesión en la base de datos.

- **RF2.3:** Empezar un nuevo juego.

1. En el panel de gestión de la sesión se seleccionará la opción “Nuevo juego”, y se seleccionará cuál en caso de que haya más de uno.
2. Se lleva a cabo el juego, en el que el sistema irá mostrando los ejercicios de forma dinámica. Esta parte se detalla en “Gestión del juego”.
3. Al finalizar el juego se vuelve a la pantalla de gestión de la sesión, desde la cual se podrá comenzar un nuevo juego, ver los resultados de juegos anteriores o finalizar la sesión.

3.2.3 Gestión de resultados

En este apartado se van a analizar los requisitos funcionales relacionados con la gestión de resultados, que incluye: captura de resultados de los juegos, almacenamiento en la base de datos y presentación al usuario. Estos requisitos son:

- **RF3.1:** Almacenar los datos recogidos durante el desarrollo del juego

1. Los datos del juego que se almacenarán son los siguientes: fecha y hora de inicio y fin, nivel de inicio y fin, total de respuestas, aciertos, fallos por omisión (el sistema no recibe respuesta) y fallos por comisión (el sistema recibe una respuesta errónea).
2. Durante el juego, en cada ejercicio se capturan los datos necesarios. En el apartado de gestión del juego se explicará que datos se capturan en cada momento.
3. Al finalizar cada juego se presentarán los resultados.
4. Se guardarán los resultados en la base de datos, asociados a la sesión dentro de la cual está el juego.

- **RF3.2:** Consultar resultados asociados a un juego de la sesión abierta.

1. En el panel de gestión de la sesión en proceso, se selecciona el juego del cual se desean consultar los resultados y la opción “Consultar resultados”.
2. Se visualizan los datos del juego en cuestión, que se describen en el RF3.4.

- **RF3.3:** Consultar resultados de juegos de sesiones finalizadas.

1. En el panel de gestión de los usuarios se selecciona la sesión a la que pertenece el juego y la opción “Consultar sesión”.

2. Se siguen los mismos pasos que en el RF3.2.
- **RF3.4:** Presentar resultados de juego al usuario. Al realizar los pasos descritos en el RF3.2 o en el RF3.3, el sistema:
 1. Recuperan de la base de datos los resultados del juego en cuestión si estos no están cargados en memoria (por ejemplo, porque el juego pertenezca a la sesión en curso).
 2. Genera las gráficas descritas en el RF3.6.
 3. Presenta los resultados por pantalla al usuario.
 - **RF3.5:** Guardar resultados del juego en la base de datos:
 1. El sistema recibe los datos capturados del juego.
 2. El sistema realiza los siguientes cálculos relacionados con las respuestas: porcentaje de aciertos, porcentaje de fallos, porcentaje de fallos por comisión respecto al total de respuestas y respecto al total de fallos y finalmente porcentaje de fallos por comisión respecto al total de respuestas y respecto al total de fallos.
 3. El sistema realiza los siguientes cálculos relacionados con los tiempos de respuesta: tiempo medio de respuesta, tiempo medio de respuesta para acierto, fallo, fallo por comisión y fallo por omisión.
 4. El sistema almacena todos estos datos junto con los resultados capturados del juego en la base de datos.
 - **RF3.6:** Generación de gráficas:
 1. Se reciben los resultados del juego previamente cargados en memoria.
 2. Con los porcentajes de aciertos, fallos por omisión y fallos por comisión se genera un gráfico circular con los siguientes colores respectivamente: verde, amarillo, rojo.
 3. Con los tiempos medios de respuesta se genera una gráfica de puntos en la que se indica además si la respuesta fue un acierto, fallo por omisión o por comisión con los mismos colores que se especifican en el punto anterior.
 4. Se presentan los resultados por pantalla al usuario.

3.2.4 Gestión del juego

En este apartado se van a exponer los requisitos funcionales correspondientes a la gestión de los juegos que incluye la aplicación. En principio se va a implementar un solo tipo de juego, pero se deja la puerta abierta a futuras mejoras orientadas a la inclusión de nuevos juegos. De hecho, el diseño se hará teniendo esto en cuenta, con la intención de que la aplicación sea lo más escalable posible.

Los requisitos funcionales correspondientes al juego explican el desarrollo del mismo, y son:

- **RF4.1:** Cargar juego.
 1. El sistema recibirá el juego que tiene que cargar.

2. El sistema recuperará los datos del juego almacenados, que incluye, al menos: configuración de los niveles del juego y ruta de alojamiento de datos propios del juego (por ejemplo imágenes).
- **RF4.2:** Iniciar juego asociado a una sesión.
 1. Se llevan a cabo los pasos descritos en el RF2.3 para iniciar un juego desde una sesión abierta.
 2. Se creará un nuevo juego a nivel de datos en el que almacenar los resultados de cada ejercicio durante el desarrollo del juego, tal y como se describe en el RF4.3.
 3. Al finalizar el juego se realizan los pasos descritos en el RF3.5 para calcular el resto de resultados y guardarlos en la base de datos.
 - **RF4.3:** Guardar los resultados del ejercicio.
 1. En el momento que se plantea el ejercicio al usuario se empieza a contar el tiempo transcurrido hasta que se recibe una respuesta.
 2. El sistema recibe una respuesta al ejercicio por parte del usuario. Para el tiempo y procesa la respuesta.
 3. Durante el procesamiento de la respuesta se modifican los contadores de respuesta y el que corresponda dependiendo del resultado de la misma: número de aciertos, de fallos por comisión o de fallos por omisión.

3.2.5 Descripción del juego

En este apartado se va a describir el juego a implementar. Se ha denominado “Juego de Coincidencia Simbólica” (a partir de ahora JCS) debido a que el propósito de cada uno de los ejercicios que lo compone es indicar si las dos imágenes que se presentan coinciden o no atendiendo al criterio vigente en ese momento.

Los posibles criterios de coincidencia de las imágenes son tres: Color, forma o forma y color. El criterio irá cambiando cada 8 aciertos (no tienen que ser seguidos) de forma aleatoria a cualquiera de los otros dos siguiendo la orientación recibida al respecto por parte del psicólogo experto. Dependiendo del nivel en el que se encuentre se informará al usuario del cambio o no.

Se definen 10 niveles para este juego, en cada uno de los cuales las variables toman un valor que marcan la dificultad del nivel. Se definen en la tabla 3.1, donde:

- **MCP:** Mostrar Criterio en Pantalla permanentemente. En un cuadro de texto en la pantalla se muestra el criterio actual.
- **ACC:** Aviso de Cambio de Criterio al usuario mediante un mensaje en pantalla.
- **OIT:** Ocultar Imágenes en un Tiempo determinado. Se trata de mostrar las imágenes y pasados ciertos segundos ocultarlas. Esto fuerza al usuario a prestar una atención más constante por una parte, y a la vez contar con más tiempo para tomar la decisión.
- **LTR:** Límite de Tiempo para la Respuesta. Si no se recibe respuesta en el tiempo máximo establecido se produce un fallo por omisión.

Cuadro 3.1: Niveles definidos para el JCS

Nivel	MCP	ACC	OIT	LTR
1	Si	Si	10	10
2	Si	Si	5	5
3	Si	No	10	10
4	Si	No	5	10
5	Si	No	5	5
6	No	No	5	5
7	No	No	3	5
8	No	No	2	5
9	No	No	3	3
10	No	No	2	2

Como se puede observar en la tabla, a medida que se sube de nivel aumenta la complejidad del juego mediante la combinación de las variables arriba explicadas. Tanto la combinación de los parámetros de los niveles como los criterios para subir de nivel han sido establecidos según la orientación recibida por parte del psicólogo experto.

La cantidad de pacientes o jugadores que va a tener la aplicación es muy alta, y se requería que fuese capaz de adaptarse por sí misma de la mejor forma posible a cada uno de ellos para cumplir su objetivo: mejorar el proceso de atención.

Por esto, lo ideal es que la aplicación presente el nivel de dificultad máximo al que el usuario es capaz de jugar sin equivocarse más que lo mínimo. De esta forma, al permanecer jugando durante periodos de tiempo cada vez más largos, los errores que se produzcan no se podrán achacar a inaptitud para resolver la tarea planteada, sino en dificultad para mantener la atención en la misma.

Siguiendo este planteamiento, se hace una primera división:

- **Niveles 1-5:** Presentan en pantalla una pauta o clave que determina cuál es el criterio que se debe comparar. Estos niveles son adecuados para los pacientes que serían incapaces de llegar a la conclusión de qué criterio se usa mediante el método de prueba y error, normalmente niños de hasta 7 años, o mayores con otros trastornos de las funciones ejecutivas. También pueden servir estos niveles como paso previo con el que familiarizarse con el sistema de juego para cualquier principiante independientemente de su edad o capacidad.
- **Niveles 6-10:** No presenta la clave, por lo que es necesario un proceso hipotético-deductivo en el que evaluar la propia actuación (estar seguro de que no ha sido un error propio, lanzar hipótesis de a qué criterio ha podido cambiar y comprobarlo). Si se maneja éste sin dificultad, podrá resolver al primer o segundo intento cada cambio de regla.

El paso entre estos dos grupos de niveles depende del criterio del terapeuta, al ser un salto cualitativo grande que depende por completo del paciente y de más factores de los que la aplicación podría manejar actualmente.

Dentro de los niveles 1-5 se parte del nivel 2, porque se espera que una amplia mayoría sea capaz de resolverlo en el tiempo que se da (5 segundos). En los casos particulares que esto no

sea así (y por lo tanto se cometan errores de omisión por falta de tiempo), se amplía éste al doble bajando al nivel 1. En caso de no tener problema en ejecutar este nivel, y al acumular suficientes aciertos como para comprobar que efectivamente se entiende y domina la dinámica, se elimina el aviso de cambio de criterio (Nvl 3) y se reducen los tiempos de presentación y respuesta progresivamente, hasta detenerse en el nivel que le corresponda por capacidad.

Esto último es también lo que va a suceder, entre los niveles 6 y 10, se va a ir subiendo el nivel cuando el jugador vaya respondiendo a todas las decisiones sin agotar el límite de tiempo (5 resultados seguidos sin errores de omisión), demostrando que periodos tan largos son innecesarios para tomar la decisión y que por lo tanto las veces que tarde más que el tiempo previsto en el nivel al que llega, será por desatención y no por capacidad, siguiendo el objetivo base de todo este sistema de niveles.

Además de lo expuesto en la figura 3.1 el terapeuta puede forzar cualquier subida o cualquier bajada en el momento que lo desee durante el desarrollo del juego. Al final del juego se guarda el nivel alcanzado por el usuario con el objetivo de que la siguiente vez que juegue continúe en ese mismo nivel, salvo que el terapeuta indique lo contrario.

En otras palabras, la primera vez que un usuario juega (es decir, no existen datos almacenados de juegos previos para el usuario en cuestión) el terapeuta elegirá si parte del nivel 2 o del nivel 6. A partir de entonces es el sistema el que se encarga de subir o bajar al usuario de acuerdo a las reglas de la figura 3.1.

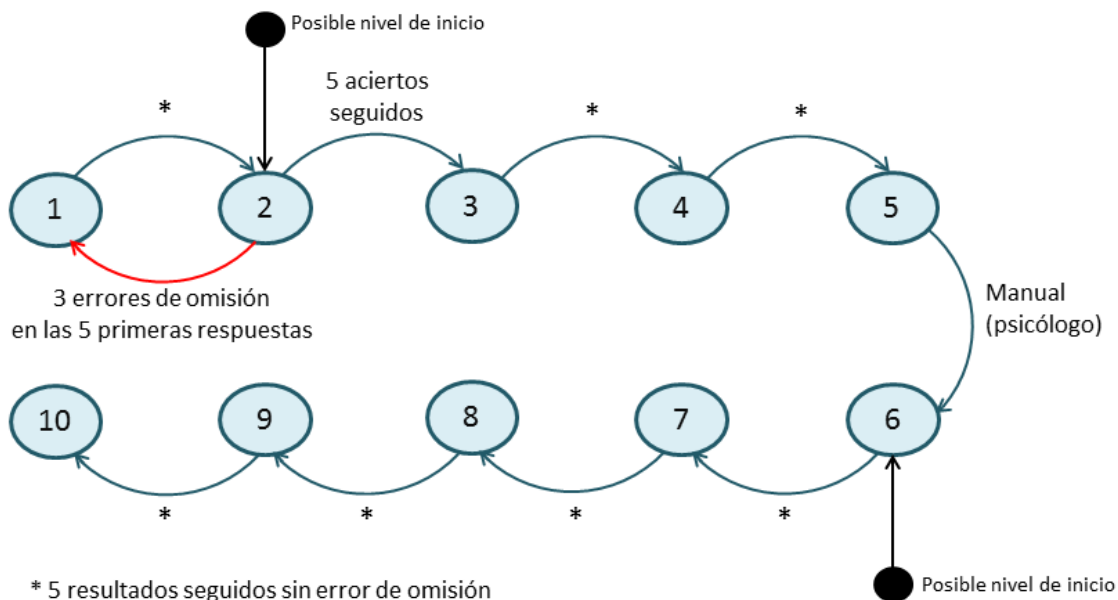


Figura 3.1: Diagrama explicativo de los criterios de cambio de nivel

3.3 Requisitos no funcionales

En este apartado se expondrán los requisitos no funcionales de la aplicación, es decir, aquellos que hacen referencia a aspectos de la misma no relacionados con la funcionalidad o el comportamiento.

Se definen los siguientes requisitos no funcionales:

- **RNF1:** La aplicación será ejecutable sobre un dispositivo específico. En este caso se realizarán las pruebas con la mesa multicontacto *Samsung SUR40* y el sistema operativo Windows.
- **RNF2:** La aplicación se diseñará de forma que sea fácilmente escalable y que en un futuro se puedan añadir nuevos juegos, para lo cual se aproveche al máximo el código ya existente.
- **RNF3:** En línea con el RNF2, se realizará un diseño modular para facilitar la incorporación futura de nueva funcionalidad.
- **RNF4:** En cuanto a la interfaz gráfica, a la hora de diseñarla se tendrán en cuenta las características de las personas con TDAH y mediante la lectura de documentación específica y la ayuda del psicólogo especializado se pensará en una interfaz adaptada.
- **RNF5:** Para aumentar la seguridad de los datos de los pacientes se requerirá una contraseña de entrada en la aplicación. Se plantea como mejora futura el cifrado de los archivos que almacenan la base de datos.
- **RNF6:** La aplicación tendrá un tiempo de respuesta inferior a 5 segundos en la carga de datos e inferior a 2 segundos durante el desarrollo de los juegos.
- **RNF7:** Todo el código estará debidamente organizado y comentado para facilitar su posterior mantenimiento.
- **RNF8:** Dada su constante interacción con datos, es esencial que la aplicación sea estable, para evitar corromper dichos datos.

4

Diseño de la solución

En este capítulo se va a describir todo el proceso de diseño de la aplicación partiendo del catálogo de requisitos desarrollado en el capítulo anterior. Además se justificarán debidamente las decisiones tomadas tales como el lenguaje de programación a utilizar, cómo se hará la base de datos, etc.

Si en los requisitos se trataba de establecer el “qué”, el diseño aborda el “cómo”. Es importante destacar que, al igual que los requisitos son una parte imprescindible de la aplicación ya que en ellos se basa todo el trabajo posterior, un diseño sólido y bien pensado es fundamental para garantizar el éxito de un proyecto, especialmente en equipos de trabajo formados por varias personas en los que cada miembro es responsable de desarrollar una parte.

En primer lugar se va a dar una visión general de la aplicación y su arquitectura para posteriormente detallar cada uno de los módulos que la componen y explicar la interacción entre los diferentes módulos y como se pasan los datos de unos módulos a otros. Finalmente, a modo de conclusión se expondrán las decisiones de diseño.

4.1 Arquitectura de la aplicación

Se define la arquitectura de un sistema como “la manera en la que es diseñado a partir de los requisitos expuestos en la fase anterior del proceso de desarrollo de software”.

En este apartado se va a dar una visión general de la arquitectura de la aplicación que estamos construyendo en este proyecto y una breve explicación de cuales son los módulos que la componen, para detallar cada uno de ellos en el siguiente apartado.

A continuación se presenta un diagrama con la arquitectura del sistema. Se reflejan en él los principales módulos y partes de la base de datos y su interacción.

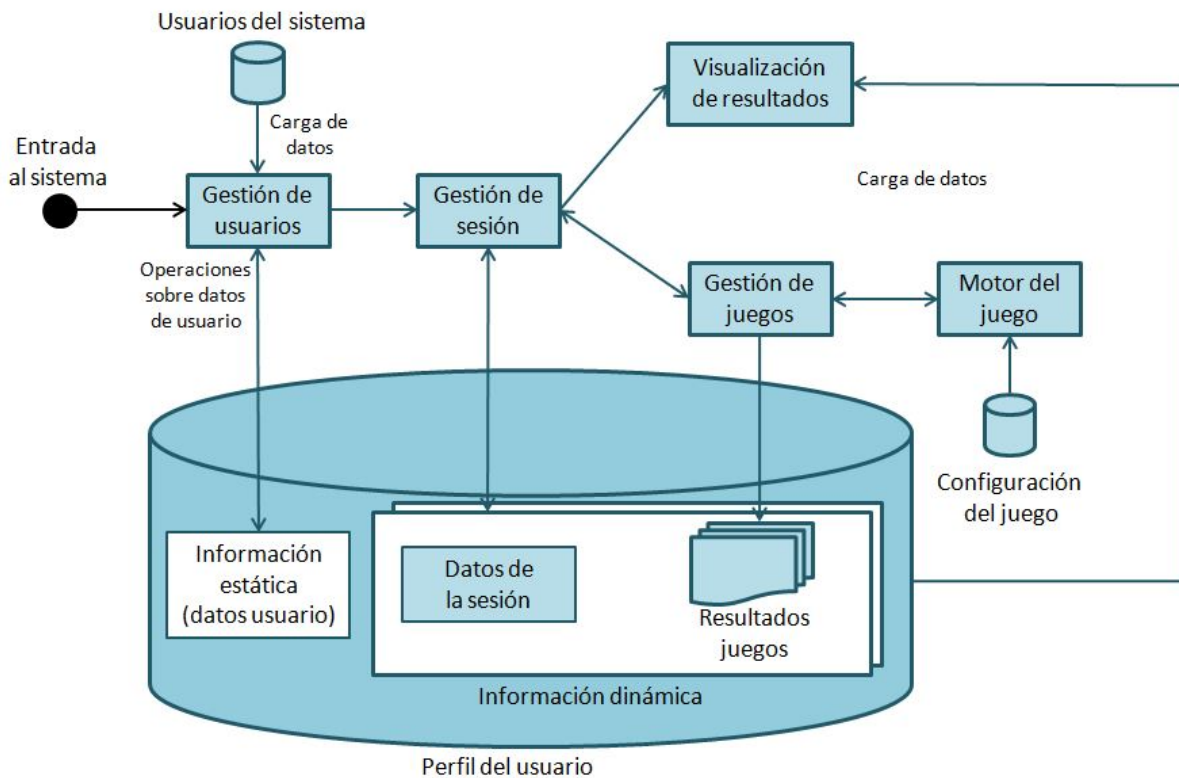


Figura 4.1: Arquitectura del sistema

Tal y como podemos ver en la figura 4.1, el sistema consta de:

- Varios **módulos** en los que se desarrollará la funcionalidad de la aplicación.
- Una **base de datos** compuesta por varias partes, cada una de las cuales almacena los datos correspondientes a uno de los módulos.
- Módulos específicos para la **comunicación** del programa con la base de datos. Si bien estos módulos no se reflejan directamente en la figura 4.1, se deduce que entran en juego en cada interacción del sistema con la base de datos.

Para entender mejor el sistema se va a exponer en forma de caso de uso lo que sería el uso normal de la aplicación en una sesión real de terapia con el paciente. En este caso de uso se relacionará cada acción sobre la aplicación con el módulo que la lleva a cabo, lo que también ayudará a entender la funcionalidad de cada uno.

1. Al entrar en el sistema, se cargarán desde la base de datos los nombres, apellidos y fotos de perfil de los usuarios registrados en el sistema, para reflejarlos en el panel de usuarios. El sistema informará también del número total de usuarios registrados en el sistema. Todas estas acciones serán llevadas a cabo por el módulo de gestión de usuarios.
2. Al elegir un usuario el sistema cargará desde la base de datos toda la información de este usuario, así como un extracto de las sesiones que se han llevado a cabo con él, nuevamente mediante el módulo de gestión de usuarios. No se cargará en este momento toda la información de las sesiones, juegos y resultados de dicho usuario para no sobrecargar la memoria. Esta información solo se cargará si el usuario quiere acceder a ella.

3. Si se desean modificar los datos del usuario cargado, el sistema interaccionará con la base de datos para sobrescribir los datos guardando los nuevos. Si se desea eliminar al usuario, el sistema ordenará a la base de datos borrar todos los asociados al usuario y se volverá al panel de usuarios. Estas acciones, denominadas en la figura 4.1 como “operaciones sobre datos del usuario” también las realiza el módulo de gestión de usuarios.
4. En este momento se pueden realizar dos acciones; o bien crear una nueva sesión o bien consultar resultados de sesiones anteriores. Se detallarán cada una de ellas en dos casos de uso distintos a continuación.

Caso de uso: **Consulta de resultados del usuario.**

1. Se seleccionará la sesión de la cual se quieren consultar los datos o bien especificar si se desea obtener un historial que presente los datos de dos o más sesiones.
2. El módulo de gestión de sesiones procesará la petición del usuario y recuperará los datos de las sesiones correspondientes.
3. A partir de los datos recuperados se generan las estadísticas y gráficas con el módulo de visualización de resultados y se presentan al usuario en una pantalla diseñada para este propósito.
4. Cuando el usuario termina de ver los resultados vuelve a los datos del usuario, a partir de dónde se pueden realizar nuevas acciones.

Caso de uso: **Creación de sesión y juegos.**

1. Al elegir la opción de crear una nueva sesión para llevar a cabo juegos con el usuario, el módulo de gestión de sesiones creará una nueva sesión asociada al usuario.
2. En el momento en el que se selecciona un juego se crea un motor específico para ese juego. El motor se encarga de todas las acciones relacionadas con el juego.
3. En primer lugar el motor cargará la configuración del juego de la base de datos así como los datos que este necesite para desarrollarse, en el caso del JCS; las imágenes que van presentando al usuario en cada ejercicio.
4. En cada ejercicio, el motor cargará los datos necesarios, recibirá la respuesta, la procesará indicando si es o no correcta y llevará a cabo todas las acciones derivadas de la respuesta, si proceden. En el caso del JCS, algunas de estas acciones son cambio de nivel y cambio de criterio. Finalmente generará un nuevo ejercicio.
5. Cuando el juego finaliza, el motor es el encargado de guardar en la base de datos los resultados del juego, asociados a la sesión a la que pertenecen, que a su vez está asociada al usuario.
6. Si el usuario desea jugar de nuevo a cualquiera de los juegos, se repetirán los pasos del 2 al 5.
7. En caso de que se desee finalizar la sesión, el módulo de gestión de sesión será el encargado de completar los datos de la sesión y guardarla en la base de datos.
8. Finalmente se volverá al panel de usuarios, devolviendo el control al módulo de gestión de usuarios.

4.2 Descripción de los módulos

En este apartado se explicarán todos los aspectos de diseño relacionados con la lógica de la aplicación, desde la formación de los identificadores de usuario hasta el funcionamiento del motor del juego.

En primer lugar es importante decidir el lenguaje de programación a utilizar. Puesto que se trata de crear una aplicación ejecutable sobre una mesa multicontacto, se han investigado métodos de programación para estos dispositivos, encontrando dos a destacar.

1. **Java+MT4J:** Desarrollar la lógica de la aplicación en lenguaje Java y la interfaz gráfica utilizando la librería MT4J [6] (“*Multi Touch for Java*”), que puede descargarse en Internet y dispone de tutoriales para el programador.
2. **Herramientas de Windows [7]:** Conjunto de herramientas, entorno y SDK proporcionado por Microsoft para la programación de aplicaciones específicas para la SUR40. El lenguaje de programación principal que utiliza es C# en la plataforma .NET.

Para el desarrollo del proyecto se ha elegido la primera opción: Java y MT4J. Se han tenido en cuenta varios aspectos para tomar esta decisión, que se justifican a continuación:

- Es **multiplataforma**, es decir, aunque el sistema operativo del dispositivo multicontacto no fuera Windows se podría ejecutar el programa. No ocurre lo mismo si se desarrolla con las herramientas específicas de Microsoft, que podrían dar problemas incluso con las actualizaciones de software.
- Se han encontrado **librerías** que dan respuesta a todos los interrogantes de programación que se plantean, como la interfaz gráfica multicontacto, la lectura y escritura de archivos XML y la generación de gráficas, que he tenido que aprender a manejar.
- En cuanto al dominio de cada uno de los lenguajes, cabe decir que a lo largo de la carrera se han adquirido los **conocimientos** necesarios para plantear un proyecto de esta magnitud en Java. En cuanto a los **entornos de programación**, de manera similar, existe una gran familiarización con el entorno Eclipse debido a las prácticas realizadas durante la carrera. El uso de herramientas y lenguajes de programación conocidos ha permitido dedicar más tiempo al resto de aspectos del proyecto.

Se necesitan las siguientes bibliotecas adicionales. Todas ellas se pueden descargar de Internet y están documentadas mediante sus respectivas interfaces de programación de aplicaciones (API).

- **MT4J:** Se utilizará para la construcción de la interfaz gráfica. De código abierto y disponible en Internet junto a ejemplos, tutoriales y guías para el programador.
- **JDOM:** Empleada para la lectura y escritura de documentos XML.
- **JFreeChart:** Se usará para la creación de gráficas en el módulo de visualización de resultados.

Por tanto aunque se utilice un lenguaje conocido para el desarrollo de la aplicación ha sido necesaria la familiarización con estas bibliotecas, especialmente con la que se construirá la interfaz gráfica.

4.2.1 Lógica general de la aplicación

En este apartado se va a dar una visión general de la lógica de la aplicación para posteriormente abordar cada módulo de forma individual entrando en los detalles que serán claves a la hora de implementar el producto.

Dado que el lenguaje de programación elegido es un lenguaje **orientado a objetos**, la forma de presentar la lógica de la aplicación es mediante un diagrama de clases. Sin embargo ya que posteriormente se va a entrar en detalle de cada uno de los módulos, en el siguiente diagrama no se incluyen los atributos ni los métodos de las clases por cuestión de espacio. La figura 4.2 presenta dicho diagrama de clases.

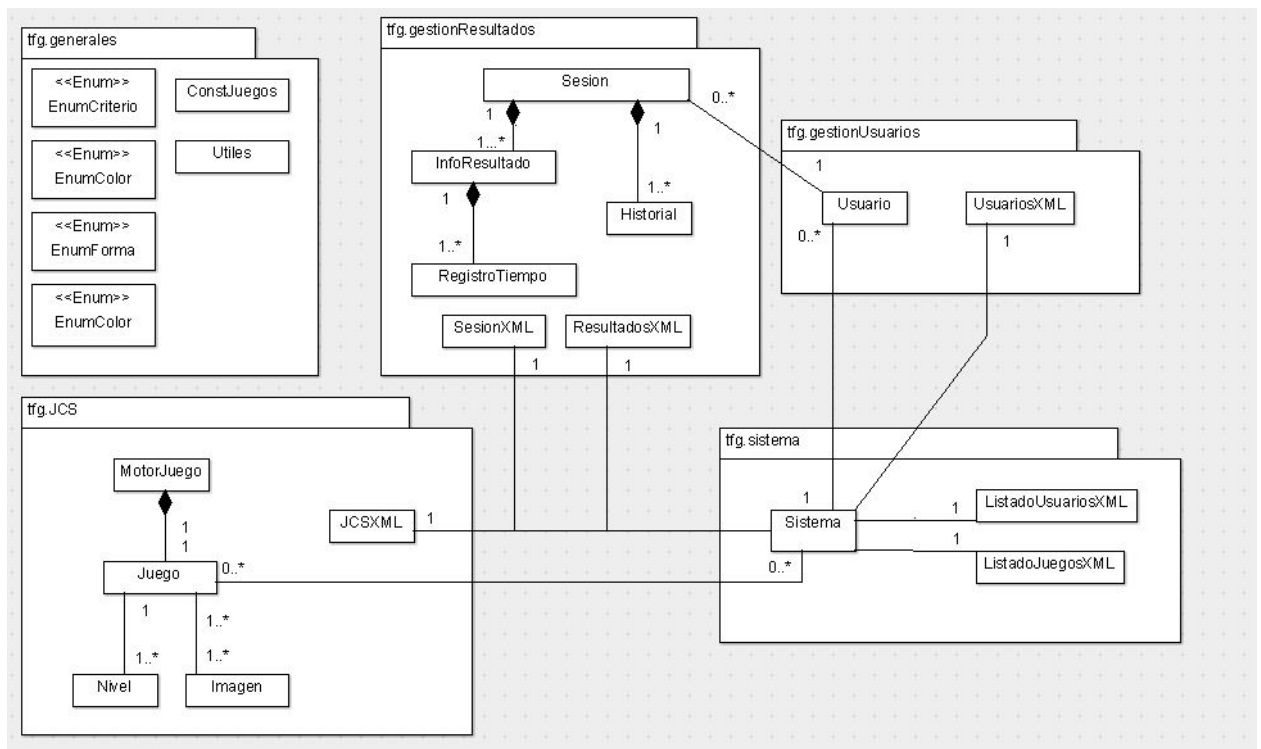


Figura 4.2: Diagrama de clases sin detallar de la aplicación

4.2.2 Lógica de los módulos

A lo largo de este apartado se van a explicar todos los detalles de cada uno de los módulos que componen la aplicación. Para ello se presentarán diagramas UML de cada módulo que incluirán todas las clases que lo constituyen, con los atributos y métodos de cada una. Asimismo, se explicarán los métodos que hayan sido implementados con una base teórica a partir de fórmulas o modelos.

Todas las tablas se adjuntan en el anexo B para no sobrecargar este apartado.

En primer lugar se va a analizar el módulo **sistema**. Este módulo contiene tres clases, como se puede ver en el diagrama de la figura 4.3.

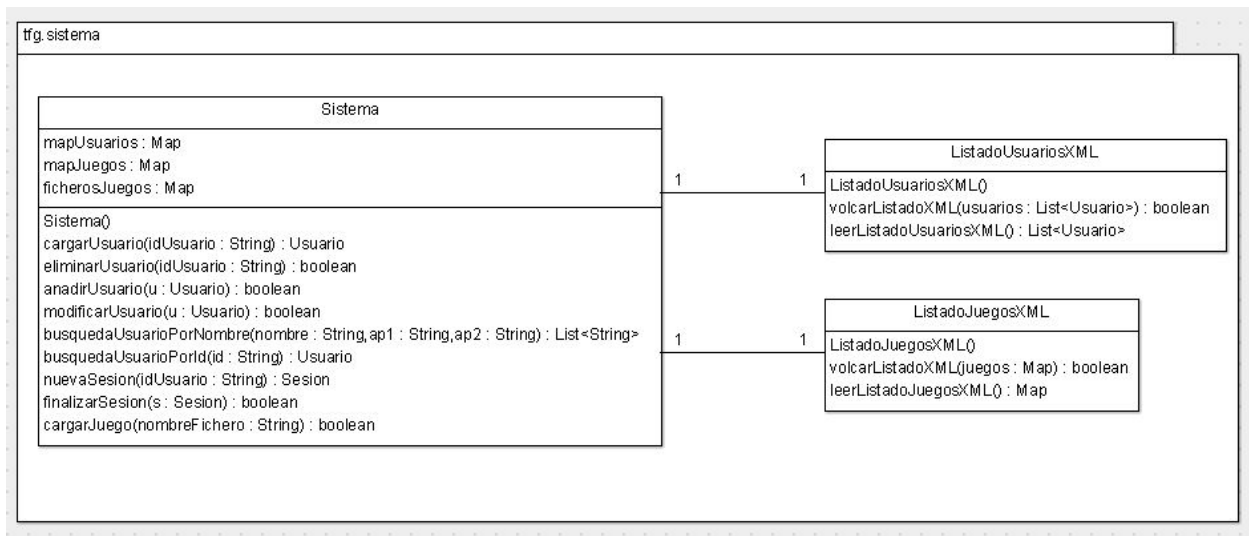


Figura 4.3: Diagrama de clases del módulo sistema

El correcto funcionamiento de este sistema es fundamental para la aplicación, ya que este sistema es el encargado de cargar todos los datos iniciales de configuración. Contiene dos clases de comunicación con ficheros XML: ListadoUsuariosXML y ListadoJuegosXML, cada una de las cuales se encarga de cargar, respectivamente el listado de usuarios del sistema y el listado de juegos disponibles.

En la tabla B.1 se presenta la descripción de los atributos de la clase Sistema.

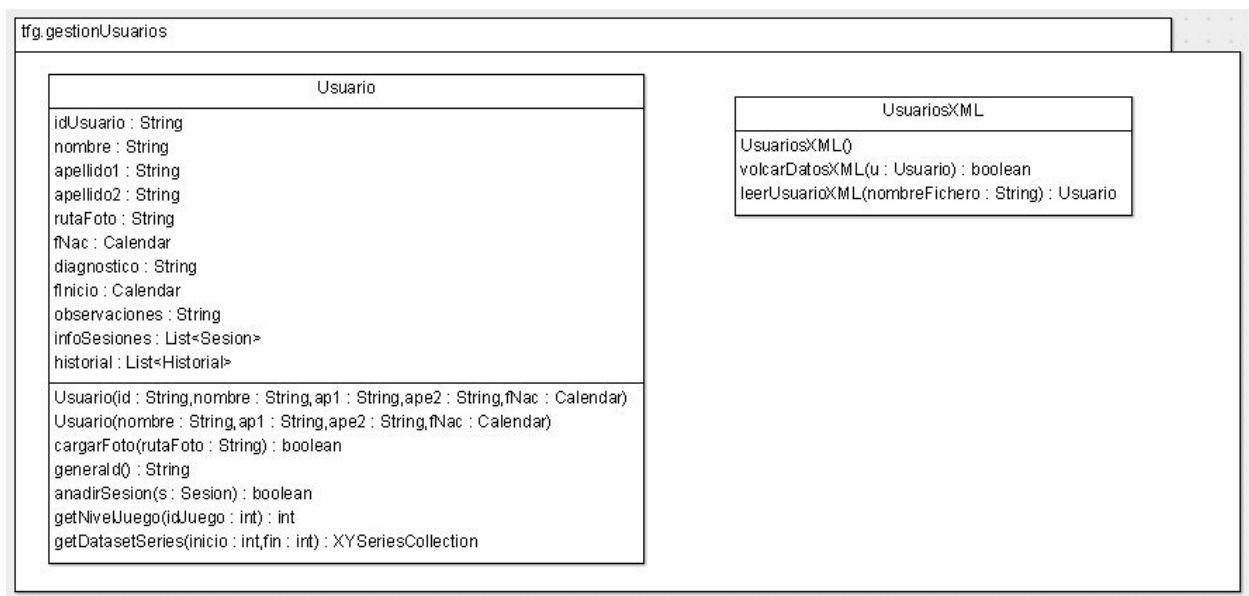


Figura 4.4: Diagrama de clases del módulo de gestión de usuarios

A continuación se presenta el módulo de **gestión de usuarios**; el que se encarga de todo el manejo de la información de los usuarios, así como de dar de alta los nuevos usuarios o eliminar alguno existente en la base de datos. En la figura 4.4 se presenta el diagrama de clases corres-

pondiente a este módulo.

Como se puede observar, este módulo consta de dos clases: la clase Usuario, que contiene todos los atributos necesarios para almacenar los datos de un usuario y la clase UsuarioXML que, como se puede deducir, es la encargada de la lectura de los archivos que contienen los datos del usuario. No se encarga de la lectura y escritura del archivo “listadoUsuarios.xml”. Como se ha comentado anteriormente este cometido lo realiza la clase ListadoUsuariosXML perteneciente al módulo del sistema.

En la tabla B.2 se realiza una descripción de los atributos de la clase Usuario.

El siguiente módulo que se analizará es el de **gestión de la sesión** junto al de **gestión de resultados**, ya que son clases con gran interacción entre ellas. Se encargan de todas las operaciones llevadas a cabo desde el momento que se crea una sesión hasta que se finaliza, incluyendo el almacenamiento de datos de la misma en los ficheros XML. En la figura 4.5 se presenta el diagrama de clases de este módulo.

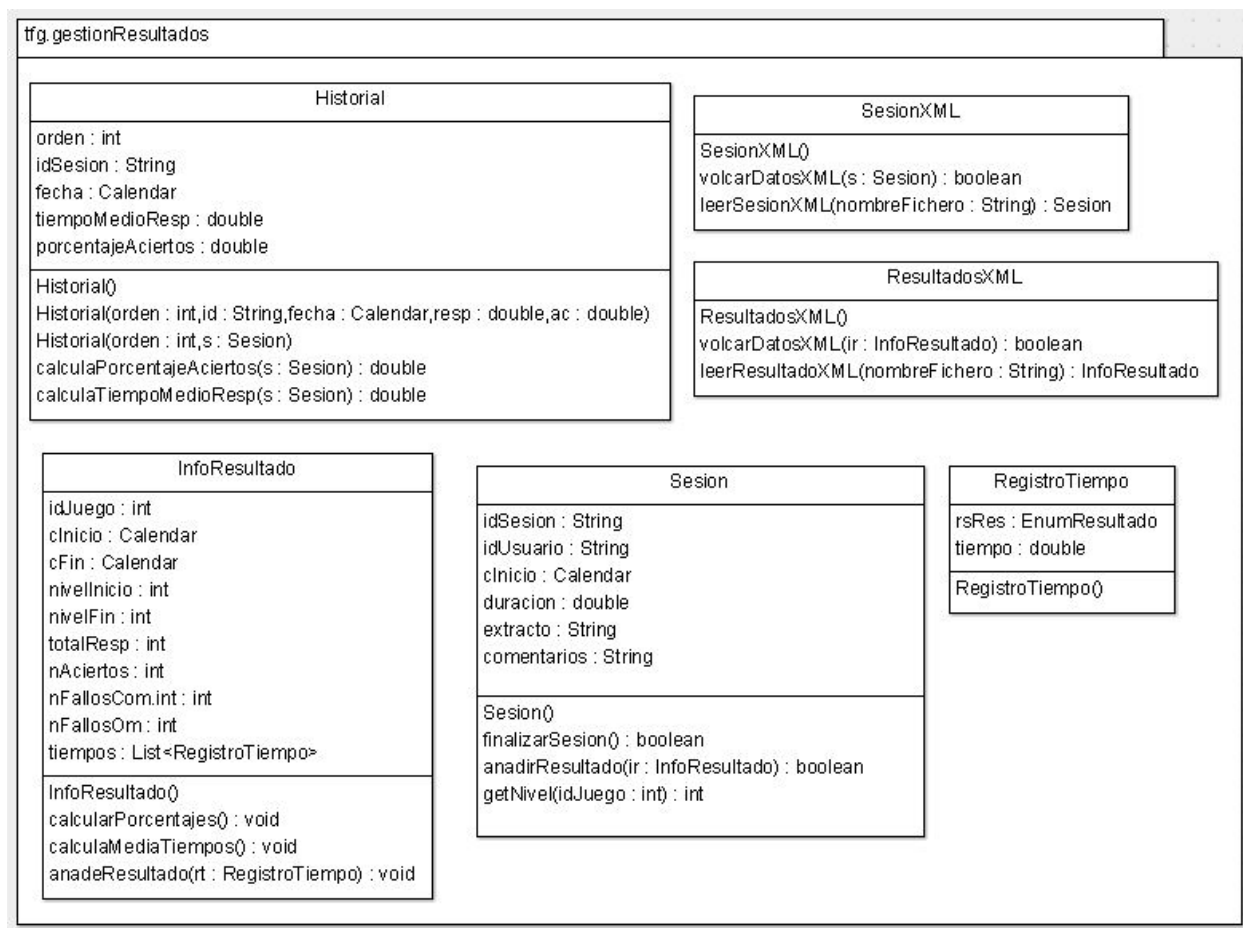


Figura 4.5: Diagrama de clases del módulo de gestión de resultados

Como se puede ver en el diagrama, hay más clases que en el resto de módulos. Se debe en parte a la creación de algunas clases auxiliares para una mejor organización de los datos. A continuación se presentan varias tablas: en la B.6 se explican los atributos de la clase Sesion, en la B.4 los de la clase InfoResultado y finalmente en la B.5 los de la clase auxiliar RegistroTiempo.

A continuación se procederá a explicar detalladamente el módulo de **gestion del JCS**. Este módulo realiza todas las operaciones necesarias para que se pueda llevar a cabo el juego, desde la generación de imágenes a la comprobación de la corrección de la respuesta y construcción del objeto con los resultados del juego llevado a cabo.

En la figura 4.6 se puede ver el diagrama de clases de este módulo, aunque en este caso es más importante la explicación posterior de, entre otros, toda la base teórica que lleva por detrás para hacerlo útil.

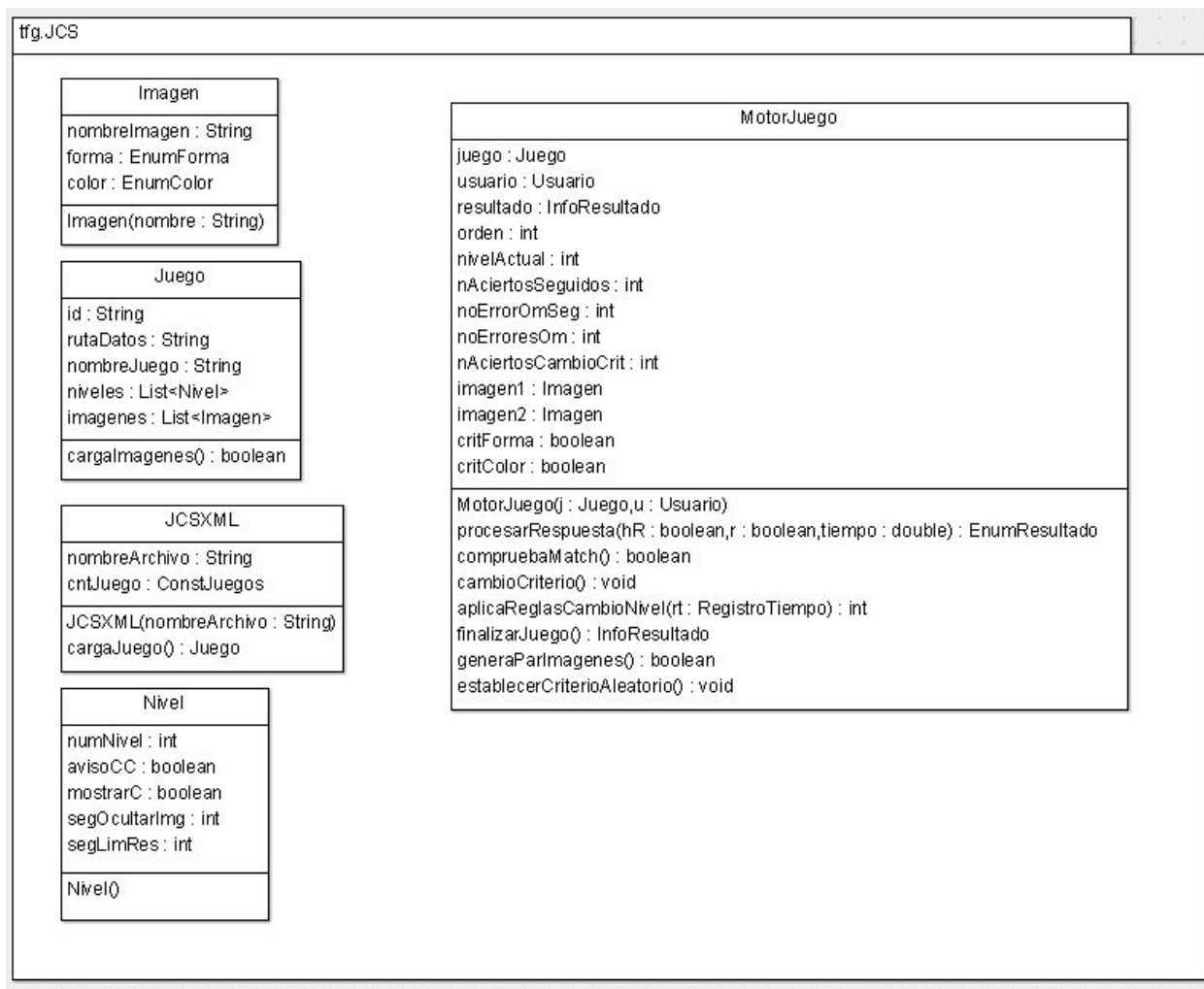


Figura 4.6: Diagrama de clases del módulo de gestión del juego

Una vez descritas de forma general las clases que componen el juego, se procederá a una explicación detallada de los atributos que componen las clases Juego, MotorJuego y Nivel en las tablas B.7, B.9 y B.8 respectivamente.

A continuación, dado que con la clase MotorJuego se implementa toda la lógica del juego, se van a explicar los métodos junto con la base teórica que tendrá cada uno de ellos:

- **MotorJuego:** Constructor de la clase, que recibe dos argumentos: un objeto juego y un usuario. El objeto juego contiene las reglas y niveles del juego que se va a llevar a cabo, y

el usuario contiene toda la información de un usuario del sistema. Se crea un nuevo motor con el juego al que se le asocia el usuario.

- **procesarRespuesta:** Recibe dos *boolean* que indican si se ha producido la respuesta y un *double* que indica el tiempo que ha tardado en darse ésta. Devuelve un objeto que contiene los datos del resultado y además realiza las llamadas necesarias a otros métodos para comprobar si la respuesta es correcta, si se tiene que producir un cambio de nivel o de criterio y finalmente guarda todos estos datos en el registro del juego que se va construyendo a lo largo de su desarrollo.
- **compruebaMatch:** Comprueba si la respuesta es correcta consultando las imágenes actuales y el criterio. Devuelve un *boolean* que indica si la respuesta del usuario es correcta.
- **cambioCriterio:** Este método realiza el cambio de criterio cuando es invocado. Cambia a cualquiera de los otros dos criterios aleatoriamente. Se implementará de forma que cada uno tenga un 50 % de probabilidad de ser el nuevo criterio.
- **aplicaReglasCambioNivel:** Aplica las reglas de cambio de nivel consultando los acumuladores de la clase MotorJuego que se van actualizando en cada ejercicio. Las reglas de cambio de nivel se detallaron anteriormente en la figura 3.1.
- **finalizarJuego:** A partir de los datos que se han ido recopilando se calculan los porcentajes y las medias de tiempos. Finalmente se fija la fecha y hora de fin del juego para posteriormente calcular su duración.
- **generaParImagenes:** Este es el método que genera un nuevo par de imágenes para el siguiente ejercicio. La primera imagen siempre se selecciona aleatoriamente entre todas las que hay y la segunda imagen se genera dependiendo del criterio actual se generan de la siguiente forma:
 1. Si el criterio es **forma y color**, existe una probabilidad del 50 % de que la segunda imagen sea la misma que la primera (para que coincidan) y una probabilidad del 50 % de que solo coincida en forma o en color.
 2. Cuando el criterio es **forma o color**, la segunda imagen tendrá un 50 % de posibilidades de coincidir y un 50 % de posibilidades de no coincidir.

4.3 Descripción de la base de datos

En este apartado trata todos los aspectos de diseño relacionados con la base de datos. En primer lugar se especificará la tecnología a usar para su desarrollo, justificando la elección y posteriormente se detallará cada una de las partes que compondrán la base de datos y qué se datos almacenará cada una de ellas.

A la hora de elegir como implementar la base de datos surgen dos opciones: base de datos **relacional** utilizando el lenguaje SQL y *Javabeans* o almacenamiento en **ficheros** binarios o XML. Nos decantaremos por el almacenamiento en ficheros XML por, entre otros, los siguientes motivos:

- Proporciona una flexibilidad total en cuanto a la organización y nomenclatura de los ficheros. Se organizarán los ficheros en carpetas de manera que al recuperar datos desde

un módulo se accederá a la carpeta en la que se encuentran los datos correspondientes a dicho módulo.

- Dado que el lenguaje que se usará para el desarrollo es Java y existen librerías específicas para la lectura y escritura de archivos XML, se utilizarán estas librerías para desarrollar los módulos específicos de comunicación con la base de datos.
- Se aísla totalmente la base de datos de la aplicación. Por ejemplo, si en un futuro se empleara otro lenguaje de programación para implementar nuevas partes de la aplicación solo habría que crear módulos específicos de comunicación. Si se utilizaran *Javabeans* habría que replantear la base de datos o hacer la nueva parte de una forma diferente, por lo que la aplicación sería menos escalable.
- Al ser un formato más abierto se facilita el procesamiento de los datos por parte de posibles *scripts* futuros.

Como ya se ha comentado, se van a organizar los ficheros que almacenan los datos de la aplicación en carpetas, de esta forma cada módulo específico de comunicación programa-base de datos accederá a la carpeta en la que se encuentran únicamente los ficheros cuyos datos necesita leer.

A continuación se listan las carpetas y se da una breve descripción de los ficheros que contienen:

- **Configuración de los juegos:** En esta carpeta se guardarán dos tipos de archivos: un listado de los juegos que es capaz de cargar la aplicación y un archivo por cada juego, que contiene la configuración y niveles de este.
- **Datos de resultados:** Contiene un archivo por cada resultado de cada juego de cualquiera de los usuarios del sistema.
- **Datos de sesiones:** Contiene un archivo por cada sesión de cualquiera de los usuarios.
- **Datos de usuarios:** Contiene un archivo en el que se listan todos los usuarios del sistema con nombre, apellidos y foto de cada uno y un archivo para cada uno de los usuarios en el que se guardan todos sus datos.
- **Fotos de usuarios:** Contiene un archivo de imagen para cada usuario que ha guardado una foto asociada a su perfil.
- **Imágenes del JCS:** Contiene las imágenes que necesita el juego para funcionar.

También es importante comentar un último aspecto relacionado con los ficheros de la base de datos. Como se puede ver en el párrafo anterior, se separan ficheros de sesión, de usuario y de resultados, así pues, necesitamos tener un sistema de nomenclatura de archivos que cumpla dos propósitos. Por un lado evitar que se repitan nombres para evitar confusiones y por otro lado asignar nombres con los que, en caso de necesitarlo, podamos identificar a que usuario o sesión pertenecen sin necesidad de guardarlo como dato en el propio archivo.

Así pues, la forma de generar los nombres de los ficheros es combinar partes de los datos del usuario con fechas y horas de la siguiente forma:

- Para los archivos de configuración de juegos, se tiene el archivo listado llamado “*listado-Juegos.xml*”, en el cual se asignará un identificador secuencial a cada uno de los juegos que se vayan registrando en el sistema. El nombre del fichero de configuración de dicho juego será *juego+identificador.xml*.
- Para formar el nombre de los ficheros que contienen los datos de resultados se utiliza la siguiente combinación: *idJuego_fecha_hora.xml*. A la fecha y hora se les quitan todos los caracteres que no sean números para evitar problemas con el sistema de nombres del sistema operativo, en este caso Windows.
- Los nombres de los ficheros que contienen los datos de sesión se formarán de la siguiente manera: *idUsuario_fechaHora.xml*. Nuevamente se eliminará todo carácter de la fecha y la hora. La formación del id del usuario se realiza poniendo las dos primeras letras del nombre y de cada uno de los dos apellidos seguidas de la fecha de nacimiento sin más caracteres que los números. Por ejemplo, Lucía Gómez López cuya fecha de nacimiento es el 1 de Mayo de 1991 tendría el identificador LUGOLO151991.
- Para la lista de los datos de los usuarios utilizaremos un fichero llamado “*datosUsuarios.xml*” y para cada uno de los ficheros que contienen los datos de los usuarios se utilizará la siguiente nomenclatura: *idUsuario.xml*.
- Para las fotos de los usuarios se utilizará *idUsuario.ext* donde extensión será la extensión del archivo de imagen.
- Finalmente, los nombres de las imágenes del JCS tendrán la siguiente combinación: *forma_color.png*. De esta forma, cuando se cargan las imágenes se da al sistema toda la información que necesita para posteriormente comprobar la coincidencia entre dos imágenes.

4.4 Interfaz gráfica

Este apartado se va a dedicar para hablar de la interfaz gráfica, las ideas de las que se parte a la hora de diseñarla y presentación de los prototipos de las pantallas en los que posteriormente se basará la implementación de las mismas.

Es importante tener en cuenta que al tratarse de una interfaz para usuarios con características especiales será necesaria la colaboración del psicólogo especialista para diseñar una interfaz útil y cómoda para ellos.

En el anexo del artículo [8] se proporciona una lista de consejos o directrices a seguir al diseñar interfaces para usuarios con necesidades especiales. A continuación se listan algunas de las que se han seguido en el proyecto.

- En cuanto al contenido:
 1. Utilizar ejemplos prácticos.
 2. Utilizar frases cortas.
 3. No utilizar palabras de otros idiomas.
- En cuanto a los elementos de la interfaz:

1. No utilizar imágenes como fondo del texto.
2. Utilizar colores para diferenciar las formas.
3. No sobrecargar la pantalla con demasiada información.

Para la implementación de la interfaz gráfica se utilizará la librería de código abierto MT4J, de la que podemos encontrar información, ejemplos y guías de programación en [6].

A continuación se van a presentar los prototipos de algunas de las pantallas más importantes de la aplicación acompañado de una breve explicación del contenido que se muestra en cada una de ellas.

La primera pantalla que se ve al iniciar la aplicación es el panel del usuarios de la figura 4.7. En esta pantalla se presentan los usuarios del sistema de una forma sencilla, con la intención de que sea el propio paciente el que localice su perfil. Para ello, además del nombre se muestra la imagen de cada usuario. El esquema de la pantalla es el siguiente:

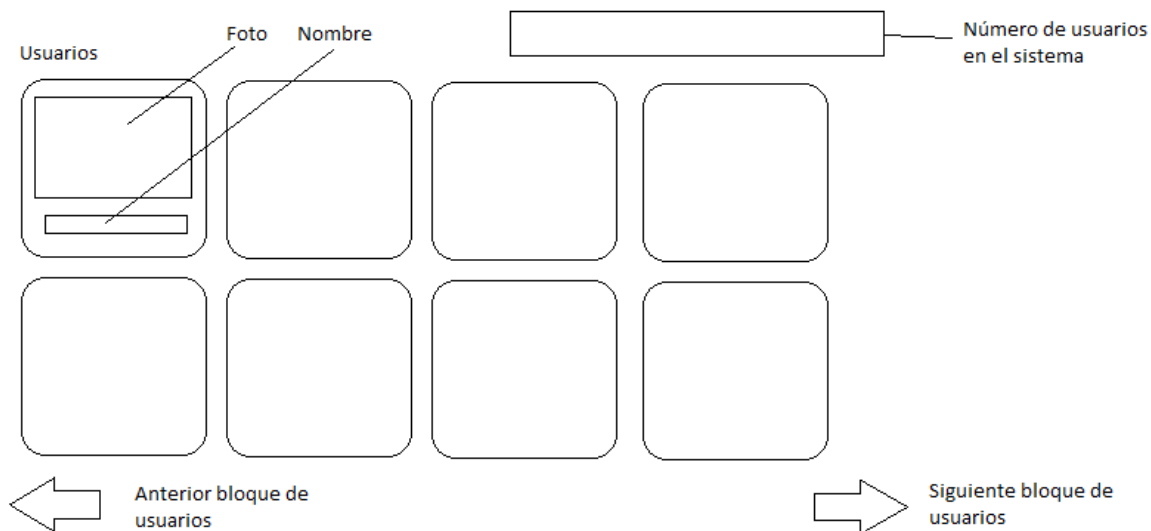


Figura 4.7: Esquema de la pantalla panel de usuarios

Una vez seleccionado el usuario se pasa a la pantalla en la que se muestra la información del usuario y de las últimas sesiones llevadas a cabo por éste y las posibles opciones en ese momento: consultar la información almacenada o bien iniciar una nueva sesión.

En este punto hay dos posibilidades, o bien consultar los datos, en cuyo caso se llevaría al usuario a la pantalla de visualización de información, o pasar a la pantalla de gestión de la sesión en la que se ofrece al usuario el catálogo de juegos disponibles para iniciar el que desee.

En el caso de este proyecto, por el momento solo hay un juego disponible (JCS). En la figura 4.8 se presenta el esquema de la pantalla del juego. Como se puede observar, el número de elementos es mínimo para favorecer la atención del paciente. Prácticamente no se da información en tiempo real para no distraer la atención y centrarla en las figuras y el criterio.

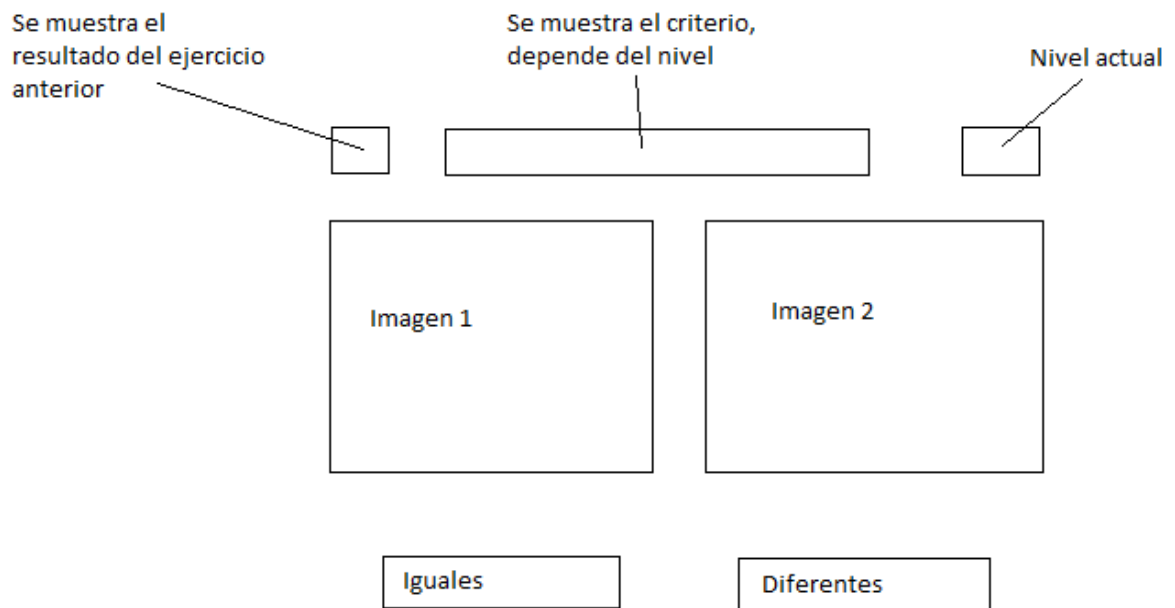


Figura 4.8: Esquema de la pantalla del juego

Finalmente comentar cuáles son las figuras y colores que se presentarán al usuario. La aplicación carga 30 figuras: seis formas (triángulo, cuadrado, círculo, rombo, estrella de cinco puntas y estrella de seis puntas) cada una en cinco colores distintos (negro, rojo, amarillo, azul y verde).

4.5 Conclusiones

Este apartado se presenta a modo de resumen del extenso capítulo de diseño de la aplicación. Se pretende así hacer una síntesis de las decisiones tomadas que afectan al siguiente capítulo, el de implementación. Se destaca por tanto, de este capítulo la siguiente información:

- Se ha elaborado un diseño concreto para la aplicación, específico para cada uno de los módulos que se van a implementar, así como la conexión entre ellos.
- Los datos se almacenarán en **archivos XML**. Estos se organizarán por carpetas y los módulos de comunicación solo accederán a la carpeta en la que se encuentran los archivos que deben leer o escribir.
- Se cargarán los datos únicamente en el momento que se necesiten, para evitar sobrecarga de la memoria y con ello una posible ralentización de la aplicación.
- Se utilizará el lenguaje **Java** para el desarrollo de la aplicación, por su amplio dominio gracias a las prácticas realizadas en la carrera y durante las prácticas en empresa. El proyecto se llevará a cabo utilizando el entorno de programación **Eclipse**, en concreto la versión “Kepler”. Nuevamente se trata de un entorno que se ha manejado a lo largo de la carrera por lo que no es desconocido.
- Se empleará la arquitectura y los métodos de la librería **MT4J** para el desarrollo de la interfaz gráfica.

- Para el desarrollo de los módulos específicos de comunicación se utilizará la librería **JDOM** y para la realización de gráficas a partir de los datos se usará **JFreeChart**.

En el siguiente capítulo se abordará la fase de implementación, se explicarán algunos de los detalles de la misma así como las dificultades que se han encontrado a lo largo del desarrollo.

5

Implementación

En este capítulo se explican todos los detalles del proyecto relativos a la fase de implementación del diseño realizado en el capítulo anterior, basado en el catálogo de requisitos expuesto en el capítulo 3. Como ya se justificó en la fase de diseño, se va a utilizar el lenguaje de programación Java y el entorno Eclipse.

A lo largo de este capítulo se tratarán los siguientes aspectos:

- **Estructura del código** desarrollado, fundamentalmente cómo se han organizado las clases en los diferentes paquetes que componen el producto final.
- Explicación detallada del funcionamiento de los **módulos de comunicación con XML**. Se comentará el funcionamiento de estas clases en particular y cómo se ha utilizado la clase JDOM.
- **Estructura** de cada uno de los **ficheros XML** que componen la base de datos. Se incluirá código con datos creados para la base de datos de pruebas de la aplicación.
- Breve explicación del **funcionamiento** de la clase **JFreeChart**, la biblioteca que se usa para generar gráficas. Se comentarán los tipos de gráficas implementadas y se mostrará código de ejemplo para la construcción de las mismas.

Además se comentarán los problemas que se han ido encontrando a lo largo del desarrollo del producto final y como se han solucionado.

5.1 Estructura del código

En este primer apartado se va a explicar la organización del código, fundamental para un mantenimiento futuro. A lo largo de la memoria se ha comentado en varias ocasiones que la intención es seguir ampliando la funcionalidad de la aplicación por cuenta propia una vez finalizado el trabajo de fin de grado.

La organización que se ha seguido es la división de las clases en paquetes nombrados como “aplicacion.paquete”. En este caso el nombre del proyecto es “tfg”, por lo que los nombres de los paquetes quedan de la forma “**tfg.nombre**”.

Dado que el análisis (capítulo 3) y diseño (capítulo 4) de la aplicación se ha realizado desde el principio agrupando la funcionalidad en módulos, la estructura de los paquetes será prácticamente la misma que la de los módulos. A continuación se listan los paquetes que constituyen la aplicación y las clases que contienen.

- **tfg.generales:** Este paquete contiene clases que se podrían dividir en tres tipos.
 1. Enumeraciones de los valores posibles para los parámetros forma, color, criterio y resultado. Como ya se comentó en la fase de diseño, el uso de estas enumeraciones, además de simplificar las comparaciones simplifica notablemente el código.
 2. Clases que contienen las constantes para los juegos y los valores fijos utilizados en la interfaz (como medidas, colores y tamaños)
 3. Clase “Utiles”, que contiene métodos cuyo uso se extiende a varios módulos y realiza acciones como transformaciones de conjuntos de datos, fechas, etc.
- **tfg.gestionResultados:** En este paquete se engloban las clases correspondientes a los módulos de sesión, resultados y visualización de información. Asimismo, se almacenan también en este paquete las clases de comunicación con XML correspondientes a sesión y resultados.
- **tfg.gestionUsuarios:** Este paquete contiene las dos clases necesarias para la gestión de usuarios del sistema: la clase Usuario y la clase de comunicación con XML de la información del usuario.
- **tfg.interfaz.componentes:** Este paquete es uno de los dos en los que se implementa la interfaz gráfica. Como el nombre indica, contiene las clases en las que se han desarrollado los componentes que se incluyen en las distintas pantallas que componen la aplicación.
- **tfg.interfaz.scenes:** Es el segundo paquete que compone la interfaz, en el están todas las clases “Scene”, cada una de las cuales implementa una pantalla o parte de ella (detallado en el apartado de interfaz mediante MT4J).
- **tfg.JCS:** Contiene todas las clases necesarias para el funcionamiento del JCS. La más importante es la que contiene la implementación del motor. También se almacena en este paquete la clase específica para la lectura de configuración y niveles de este juego.
- **tfg.pruebas:** En este paquete se han ido creando clases a lo largo del desarrollo para probar partes específicas en modo texto. Estas pruebas se detallan en el capítulo 6.
- **tfg.sistema:** Este paquete almacena clases de vital importancia para la ejecución de la aplicación, ya que son las que cargan todos los datos al arrancarla. Concretamente almacena las siguientes clases:
 1. *Sistema:* Como ya se vio en el capítulo de diseño, sus atributos almacenan toda la información de la ejecución en curso. Es el único objeto que se instancia en la escena principal de la interfaz gráfica.

2. *ListadoUsuariosXML*: Carga el listado de usuarios del sistema del archivo “listadoUsuarios.xml”.
3. *ListadoJuegosXML*: Carga la lista de los juegos disponibles en el sistema, almacenando así los archivos desde los que posteriormente se podrá cargar la configuración de cada uno de ellos.

5.2 Módulos de comunicación

En este apartado se realizará una breve descripción de los módulos de comunicación del sistema y de la biblioteca utilizada para su desarrollo.

La librería JDOM es una librería desarrollada para la lectura y escritura de archivos XML. Se puede descargar de Internet en [9] y básicamente lo que hace es crear una jerarquía de elementos igual que la del archivo XML que está leyendo o escribiendo.

La estructura de todas las clases es la misma:

- Constantes en las que se almacenan los valores de las etiquetas.
- Constructor en el que se comprueba la existencia de la ruta de la que se leerá o en la que se guardarán los nuevos archivos generados. En caso de que no exista, se crea la ruta.
- Método de volcado de datos a archivo. Este método recibe el objeto a volcar y devuelve un valor booleano que indica si se ha realizado el volcado correctamente o no.
- Método de lectura de un archivo XML. Devuelve un objeto con la información leída del archivo y *null* en caso de que se produzca algún problema durante la ejecución del método.

5.3 Estructura de los ficheros XML

En este apartado se explica la estructura de cada uno de los archivos XML que componen la base de datos mediante fragmentos de código de los mismos, con los datos que se han creado para la realización de las pruebas de la aplicación.

Como ya se ha comentado anteriormente, el archivo *listadoUsuarios.xml* contiene un listado con los usuarios del sistema. A continuación se puede ver el código correspondiente al registro de un usuario:

```
1 <usuario id="LUGOLO141991">
2   <nombre>Lucia</nombre>
3   <apellido1>Gomez</apellido1>
4   <apellido2>Lopez</apellido2>
5   <rutaFoto>\datos\fotosUsuarios\LUGOLO141991.jpg</rutaFoto>
6 </usuario>
```

Figura 5.1: Datos básicos de un usuario en *listadoUsuarios.xml*

Como se puede observar, la información almacenada en este fichero es mínima, pero suficiente para poder cargar toda la información del usuario, ya que se almacena el identificador. A partir de los datos que se almacenan en este fichero se cargará el panel principal de usuarios: nombre, apellidos y ruta de la foto relativa a la carpeta en la que se almacena el proyecto.

A continuación se presenta un archivo de usuario, en el que se guardan todos los datos del mismo. Esta información solo se carga cuando es necesario porque el usuario de la aplicación solicite consultar dichos datos.

```
1 <usuario id="LUGOLO141991">
2   <nombre>Lucia</nombre>
3   <apellido1>Gomez</apellido1>
4   <apellido2>Lopez</apellido2>
5   <rutaFoto>\datos\fotosUsuarios\LUGOLO141991.jpg</rutaFoto>
6   <fechaNac>1/9/1991</fechaNac>
7   <diagnostico>TDAH</diagnostico>
8   <fechaInicio>12/6/2013</fechaInicio>
9   <observaciones>Transtorno de deficit de atencion e hiperactividad</observaciones>
10  <sesiones>
11    <sesion idSesion="LUGOLO141991_2472013112445">
12      <duracion>0.0</duracion>
13      <fecha>24/9/2013</fecha>
14      <extracto>juegos=1.2_4</extracto>
15    </sesion>
16  </sesiones>
17  <historial>
18    <elemHistorial orden="1">
19      <idSesion>LUGOLO141991_2472013112445</idSesion>
20      <fecha>24/9/2013</fecha>
21      <aciertos>0.7142857142857143</aciertos>
22      <tiempos>2.7942857142857145</tiempos>
23    </elemHistorial>
24  </historial>
25 </usuario>
```

Figura 5.2: Información completa de un usuario

Como se puede observar, en primer lugar se encuentran los datos personales del usuario; nombre, apellidos, fecha de nacimiento, etc. A continuación están los datos suministrados por el terapeuta (fecha de inicio de la terapia, diagnóstico, observaciones...). Después se encuentra la información de las sesiones llevadas a cabo con el usuario, pero solo se muestra un extracto de éstas (que contiene el identificador de sesión a partir del cual se pueden cargar el resto de los datos del archivo de dicha sesión) y finalmente un historial de las sesiones en el que se guardan

los datos necesarios para construir las gráficas de historial de sesiones.

A continuación se puede ver el código correspondiente a un archivo en el que se almacena la información de una sesión. En primer lugar se almacenan los datos propios de la sesión como la fecha y hora de inicio, la duración y los comentarios del terapeuta. Tras toda esta información se guarda una lista de identificadores de resultados, cada uno de los cuales corresponde a un juego llevado a cabo durante la sesión. Si el usuario lo solicita se cargan los ficheros para visualizar la información que contienen.

```
1 <sesion idSesion="LUGOLO141991_2472013112445">
2   <idUsuario>LUGOLO141991</idUsuario>
3   <fechaHora>24/7/2013_11:24:45</fechaHora>
4   <duracion>3600.0</duracion>
5   <extracto>juegos=1_2_4</extracto>
6   <comentarios>Evoluciona de manera satisfactoria</comentarios>
7   <lstResultados>
8     <lst_resultado>1_2472013_112445</lst_resultado>
9   </lstResultados>
10 </sesion>
```

Figura 5.3: Información de sesión

A continuación se va a presentar el contenido de uno de los archivos en los que se almacena la información de los resultados del juego una vez este ha finalizado. Debido a su extensión por todos los datos que guarda no se muestra entero. El archivo consta de dos partes: en la primera se guardan todos los datos como tasa de aciertos, de fallos, etc. y en la segunda parte se guarda una lista de resultados, uno por cada ejercicio de los que ha constado el juego.

```
1 <resultado idResultado="1_2472013_112445">
2   <tipoJuego>1</tipoJuego>
3   <clInicio>24/7/2013_11:24:45</clInicio>
4   <clFin>24/7/2013_11:29:45</clFin>
5   <nivelInicio>2</nivelInicio>
6   <blAutomatico>true</blAutomatico>
7   <nivelFin>4</nivelFin>
8   <totalRespuestas>7</totalRespuestas>
9   <nAciertos>4</nAciertos>
10  <nFallosCom>2</nFallosCom>
11  <nFallosOm>1</nFallosOm>
```

Figura 5.4: Algunos de los campos de información de resultados del juego

En la figura anterior se muestran algunos de los campos de los que almacenan la información del juego al final del mismo, y en la siguiente un ejemplo de los datos que se guardan de cada ejercicio: resultado y tiempo de respuesta.

```
1 <lstTiempos>
2   <lst_elemento>
3     <lst_resultado>FALLOCOM</lst_resultado>
4     <lst_tiempo>4.02</lst_tiempo>
5   </lst_elemento>
6   <lst_elemento>
7     <lst_resultado>ACIERTO</lst_resultado>
8     <lst_tiempo>2.05</lst_tiempo>
9   </lst_elemento>
```

Figura 5.5: Almacenamiento de resultados del ejercicio

Por último se exponen los ficheros XML de configuración de los juegos. En primer lugar, el fichero del que se carga la lista de los juegos que maneja la aplicación, indicando el nombre del fichero del cual se podrá cargar posteriormente la configuración:

```
1 <listadoJuegos>
2   <juego>
3     <idJuego>1</idJuego>
4     <ficheroJuego>juego1.xml</ficheroJuego>
5   </juego>
6 </listadoJuegos>
```

Figura 5.6: Listado de juegos que maneja la aplicación

Finalmente, un fragmento del fichero del que se carga la configuración del juego, en este caso JCS. Consta de una ruta en la que están las imágenes y el listado de niveles del juego (solo se muestran dos).

```
1 <juego>
2   <idJuego>1</idJuego>
3   <datos>\datos\imagenesMatch</datos>
4   <listaNiveles>
5     <nivel idNivel="1">
6       <avisoCambioCriterio>True</avisoCambioCriterio>
7       <mostrarCriterio>True</mostrarCriterio>
8       <ocultarImgTiempo>10</ocultarImgTiempo>
9       <limiteTiempoResp>10</limiteTiempoResp>
10    </nivel>
11    <nivel idNivel="2">
12      <avisoCambioCriterio>True</avisoCambioCriterio>
13      <mostrarCriterio>True</mostrarCriterio>
14      <ocultarImgTiempo>5</ocultarImgTiempo>
15      <limiteTiempoResp>5</limiteTiempoResp>
16    </nivel>
```

Figura 5.7: Fichero de configuración del JCS

5.4 Interfaz gráfica mediante MT4J

En este apartado se va a dar una descripción general de los aspectos más importantes de la construcción de la interfaz gráfica. Se ha utilizado la biblioteca MT4J ya que es específica para interfaces de aplicaciones ejecutables sobre dispositivos multicontacto.

Cada una de las pantallas de la aplicación se denominan **escenas**, y cada una de ellas está formada por **componentes**. Los componentes pueden ser de diferentes tipos dependiendo del propósito para el cual hayan sido concebidos; cuadros de texto, imágenes, botones...

Una buena práctica de programación con esta biblioteca según las propias directrices de la documentación de la biblioteca es crear nuevos componentes a partir de componentes más sencillos, con sus propios métodos y configuración para que, posteriormente a la hora de incluirlos en escenas el código sea más legible y fácil de mantener.

Uno de los puntos en contra de esta librería es que no dispone de un entorno gráfico para la construcción de interfaces como por ejemplo el de *Netbeans* para Java, lo cual hace que la implementación de la interfaz se realice a mano en cada una de las clases de las escenas. Para ello el uso de una jerarquía de componentes es fundamental, ya que se colocan respecto al componente padre y no en una posición absoluta.

Finalmente exponer brevemente algunas de las pantallas mas importantes de la aplicación, la de juego y el panel de usuarios.

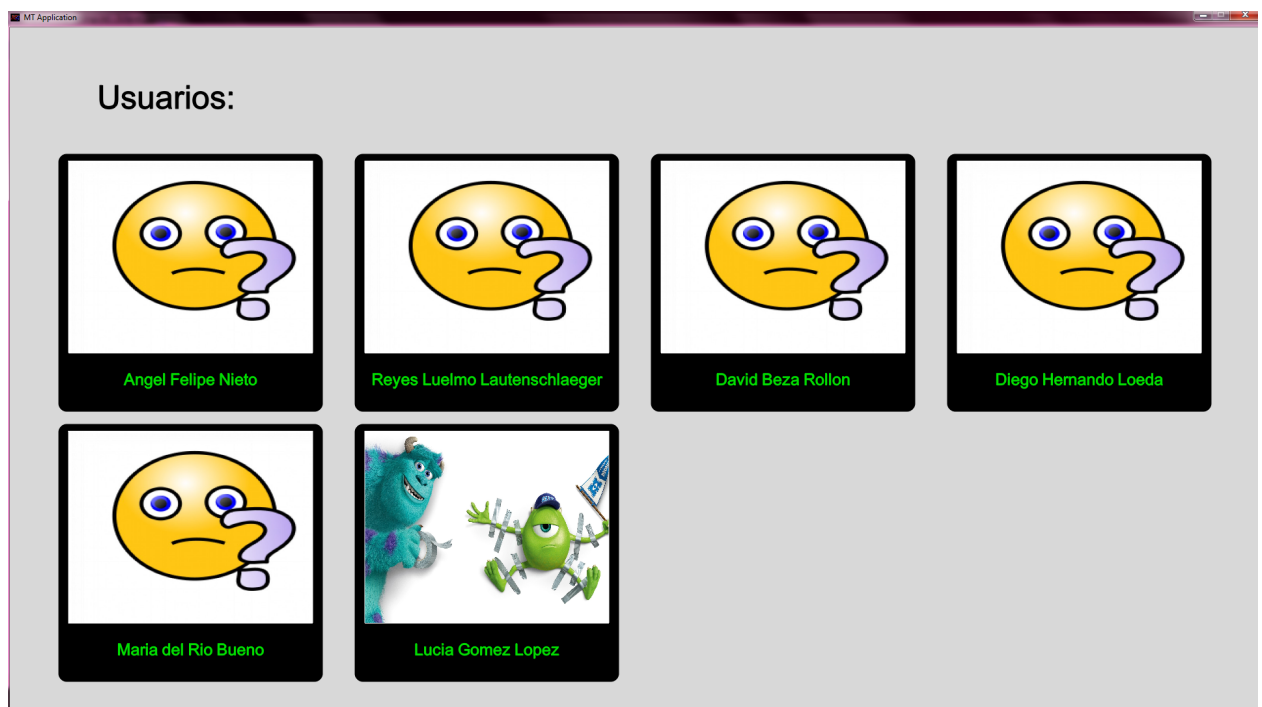


Figura 5.8: Captura de pantalla del panel de usuarios

En la figura 5.8 se muestra el panel de usuarios, que presenta todos los usuarios de la aplicación, identificados por su nombre y apellidos y una foto que puede elegir el usuario. En caso de no haber seleccionado ninguna se muestra un icono común.

Para esta escena y para otras se ha creado el componente *rectanguloUsuario*, que tiene un constructor al que se le pasa un usuario y devuelve un rectángulo como los que se pueden ver en la figura, con la imagen y el nombre del usuario. Esto hace que el código sea mucho más claro y la colocación de los elementos, mucho más cómoda.

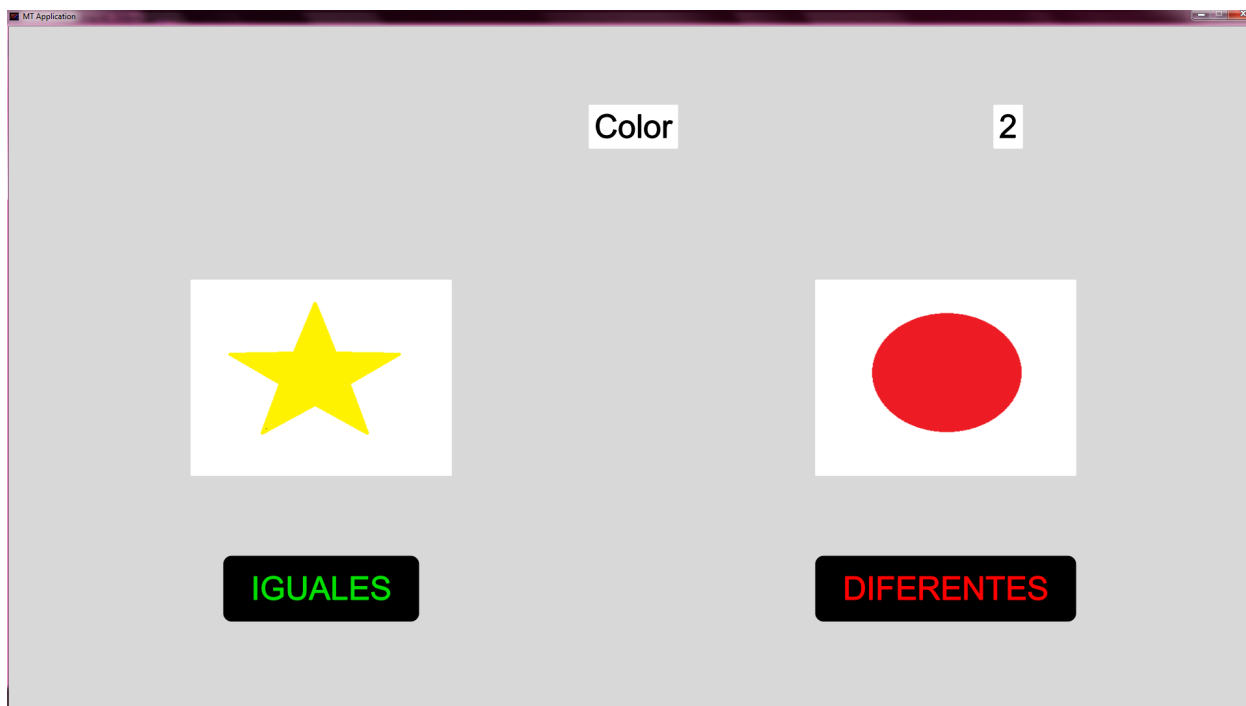


Figura 5.9: Captura de pantalla del JCS

En la figura 5.9 se muestra una captura de pantalla del juego en un momento de su ejecución. Todos los elementos que aparecen son rectángulos a los que se les ha añadido contenido y en el caso de los botones mediante los que el usuario responde al ejercicio, escuchadores de eventos que lanzan el proceso de creación de nuevos ejercicios.

Finalmente comentar que ambas capturas de pantalla se han realizado en la mesa y no en el simulador del ordenador. En el apartado de las pruebas de implantación se detallará porqué.

5.5 Conclusiones

En este apartado se van a comentar de forma breve las conclusiones y las dificultades que se han encontrado durante esta fase del desarrollo del producto.

Se han desarrollado los módulos previstos, se han establecido las conexiones entre ellos y se han representado y almacenado los datos según lo especificado en la fase de diseño detallada en el capítulo 4.

Dado el manejo del lenguaje no han surgido problemas de implementación de la lógica del juego que no hayan podido ser solucionados con una simple búsqueda en internet o consultando la API de la biblioteca.

Sin embargo, si se encontró una dificultad bastante importante a la hora de comenzar la implementación de la interfaz gráfica ya que no se conseguía hacer funcionar la biblioteca MT4J para Windows 7 de 64 bits (sistema operativo de los ordenadores en los que se ha desarrollado y en la mesa SUR40). Se acudió a las secciones de ayuda de la biblioteca y pese a ello no se consiguió solucionar hasta que a raíz de pruebas y de información encontrada en foros especializados se dio con la solución, que consistió en la sustitución de algunos archivos por otros preparados para Windows de 64 bits.

6

Implantación y pruebas

En este capítulo se van a tratar dos temas principalmente: el proceso de implantación de la aplicación en la mesa multicontacto y el plan de pruebas que se ha llevado a cabo al final y durante el desarrollo.

Ambas partes son fundamentales en el proyecto. La primera porque uno de los objetivos del trabajo es la construcción de una aplicación ejecutable sobre un dispositivo multicontacto, en este caso la Samsung SUR40. Si el funcionamiento del software sobre la mesa no fuera el correcto, uno de los objetivos fundamentales del proyecto no habría sido cumplido.

Por otro lado, el plan de pruebas es fundamental para determinar si todos los requisitos que se plantearon en el capítulo 3 se han cumplido, tanto funcionales como no funcionales.

6.1 Implantación

En este apartado se explicará brevemente el proceso de implantación de la aplicación en el dispositivo multicontacto SUR40, los problemas que han surgido y como se han solucionado.

Dado que el sistema operativo del dispositivo es Windows 7 y que el despliegue de la aplicación en la mesa no se realizó al final del proyecto sino durante el desarrollo de la interfaz gráfica, se instalaron en la mesa las mismas herramientas que en los ordenadores en los que se ha llevado a cabo el desarrollo, es decir, Java y el entorno de programación Eclipse.

La intención al reproducir las herramientas fue poder hacer correcciones sobre la marcha en la propia mesa, ya que en la documentación de la librería MT4J se avisa de que es muy posible de que la colocación de los elementos varíe ligeramente entre el simulador y el dispositivo al ser distintas las dimensiones. Se muestra un ejemplo de esto en la figura 6.1: la imagen de la izquierda corresponde a la ejecución en el simulador del ordenador mientras que la imagen de la derecha es una captura realizada en la SUR40 a pantalla completa.

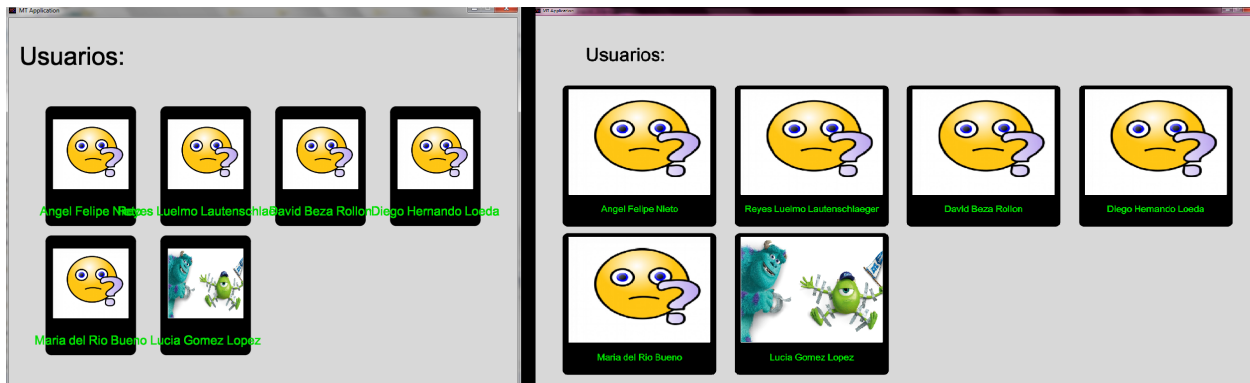


Figura 6.1: Comparativa de capturas del panel de usuarios

Como se puede observar, en la captura realizada en el ordenador los componentes de la escena se montan unos sobre otros, cosa que no ocurre en la captura realizada en la SUR40. Ha sido por tanto, una gran ventaja contar con este dispositivo a lo largo del desarrollo para corregir este tipo de cosas, y la implantación en la mesa se ha ido haciendo progresivamente según se ha ido implementando.

En cuanto a los problemas encontrados, han sido los mismos que a la hora de ejecutar MT4J sobre Windows de 64 bits y que en su momento se solucionaron gracias al apartado de “Preguntas más frecuentes- [10] de la documentación de la biblioteca.

6.2 Plan de pruebas y resultados

En este apartado se va a explicar cuales han sido las pruebas que se han realizado sobre el producto para determinar el cumplimiento de los objetivos y requisitos, tanto funcionales como no funcionales.

Se han llevado a cabo varios tipos de pruebas en diferentes momentos del proyecto:

1. **Unitarias:** Se han ido llevando a cabo a lo largo del desarrollo de la aplicación y con estas pruebas se ha ido comprobando el correcto funcionamiento de los módulos y las partes que componen.
2. **De integración:** Se han llevado a cabo al final del desarrollo de los módulos y su objetivo es comprobar la corrección de la comunicación entre ellos y el paso de datos.
3. **De sistema:** Se han realizado sobre la mesa multicontacto una vez finalizada la implementación de la aplicación. Prueban el correcto funcionamiento del producto construido una vez unido con la interfaz e implantado en la mesa.
4. **Con usuarios:** Finalmente se han realizado pruebas con usuarios reales para comprobar la utilidad de la aplicación desarrollada y ver como interactúan con ella otros usuarios distintos a los que se han ido familiarizado a lo largo de su construcción.

A continuación se expone cada tipo de prueba en un apartado, detallando qué resultados se pretenden obtener, cómo se han realizado y finalmente su resultado.

6.2.1 Pruebas unitarias

Estas son las primeras pruebas que se realizan a la aplicación pero no por ello las menos importantes. Mediante estas pruebas se va comprobando que el desarrollo es correcto en pequeños fragmentos de código.

Su objetivo principal es detectar fallos de implementación en una fase del proyecto en la que no supone un problema cambiarlo o detectarlo. Es decir, el objetivo es que a medida que se avanza y se crean módulos que engloban a otros utilizando sus métodos se pueda confiar en que están bien implementados y por tanto hacer más fácil la tarea de detectar y acotar fallos en los nuevos módulos.

Para llevar a cabo estas pruebas se han creado programas sencillos dentro del paquete **tfg.pruebas** con los que se han ido probando distintas partes de la funcionalidad de los módulos desarrollados. Cuando el funcionamiento de los programas de pruebas no ha sido el esperado se ha depurado el código, buscando el momento en el que se produce el fallo y los datos a los que afecta para poder solucionarlo.

Uno de los programas creados para este tipo de pruebas es **PruebasGraficos**, con el que se han ido realizando pruebas de la generación de gráficas con JFreeChart. En primer lugar este programa se creó para aprender a manejar los métodos de la clase para cada tipo de gráfica y posteriormente se utilizó para probar los métodos de las clases que las crean para la visualización de resultados. Uno de los ejemplos de gráficas que se obtienen con el programa es el circular que se ve en la figura 6.2. Este gráfico representa los porcentajes de aciertos y fallos en un juego utilizando colores para hacer más fácil su lectura. Al tratarse de una prueba unitaria los datos se incorporaron directamente en el código ya que no se quería mezclar esta prueba con la lectura de archivos XML para aislar problemas.

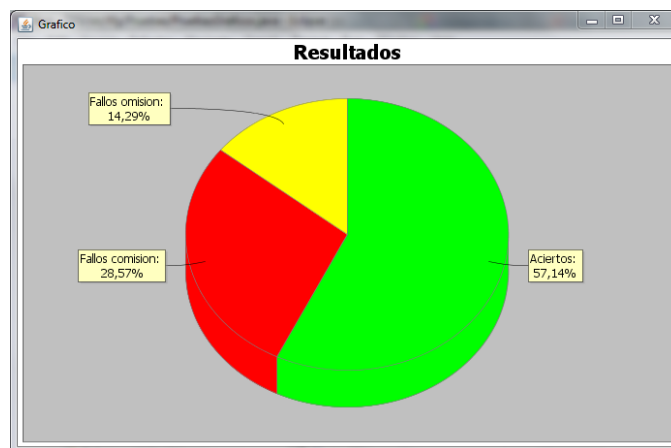


Figura 6.2: Ejemplo de gráfico circular a partir de resultados

Otros de los programas de pruebas que se han creado han sido uno para cada lector XML, tanto para el proceso de lectura como para el de escritura. En todos ellos el proceso ha sido el mismo: se simulan los datos en el programa creando un objeto con ellos y se escriben en el fichero, se leen del fichero en otro objeto y se comprueba campo a campo que el contenido de ambos objetos es exactamente el mismo.

Las pruebas de los módulos de lectura y escritura en archivos XML no han dado problemas, ya que los módulos son muy parecidos entre sí porque la estructura es la misma, por lo que una vez se solventaron los errores del primer lector, hacer los siguientes no ha sido especialmente costoso.

Por último comentar que se han realizado pruebas de todos los métodos de las clases que realizan algún tipo de cálculo sobre datos, para comprobar que estos son correctos contrastándolo con los mismos cálculos realizados a mano. En estas pruebas es donde más errores se han encontrado, la mayoría por coger los datos del atributo equivocado o por errores en acumuladores cuando se ha tratado de operaciones con listas de datos, por ejemplo, al procesar los resultados de los ejercicios.

6.2.2 Pruebas de integración

En este apartado se explicarán las principales pruebas de integración de módulos, llevadas a cabo sobre módulos completamente terminados y que utilizan métodos de otros módulos, por lo que se producen intercambios de datos.

Previo al desarrollo de estas pruebas se ha creado una versión más estable de la base de datos, con resultados simulados de sesiones y juegos. Cada vez que se han realizado pruebas de eliminación de datos se ha restablecido todo el conjunto mediante la ejecución de un programa creado para tal efecto: **CreaDatos**, que elimina todos los datos existentes y crea el nuevo conjunto, incluida escritura en ficheros. Este programa ha sido de gran utilidad ya que ha evitado tener que crear los datos manualmente, ahorrando mucho tiempo y posibles errores.

En general, el resultado de las pruebas ha sido satisfactorio, en parte porque previamente a las pruebas de integración se realizaron las unitarias a cada uno de los módulos, por lo que los errores han sido mucho más fácil de acotar y solucionar.

La prueba que más problemas generó y costó solucionar fue la de eliminación de usuarios. No por la eliminación del usuario en sí, tarea relativamente sencilla, sino por la cantidad de datos asociados que hay que eliminar en cascada, extrayendo los identificadores de los archivos de sesión y resultados del propio usuario. La forma de eliminar es de “abajo a arriba”, es decir, primero los archivos de resultados, luego los de sesión, el de los datos del usuario y finalmente eliminar al usuario del archivo en el que se listan todos los usuarios del sistema.

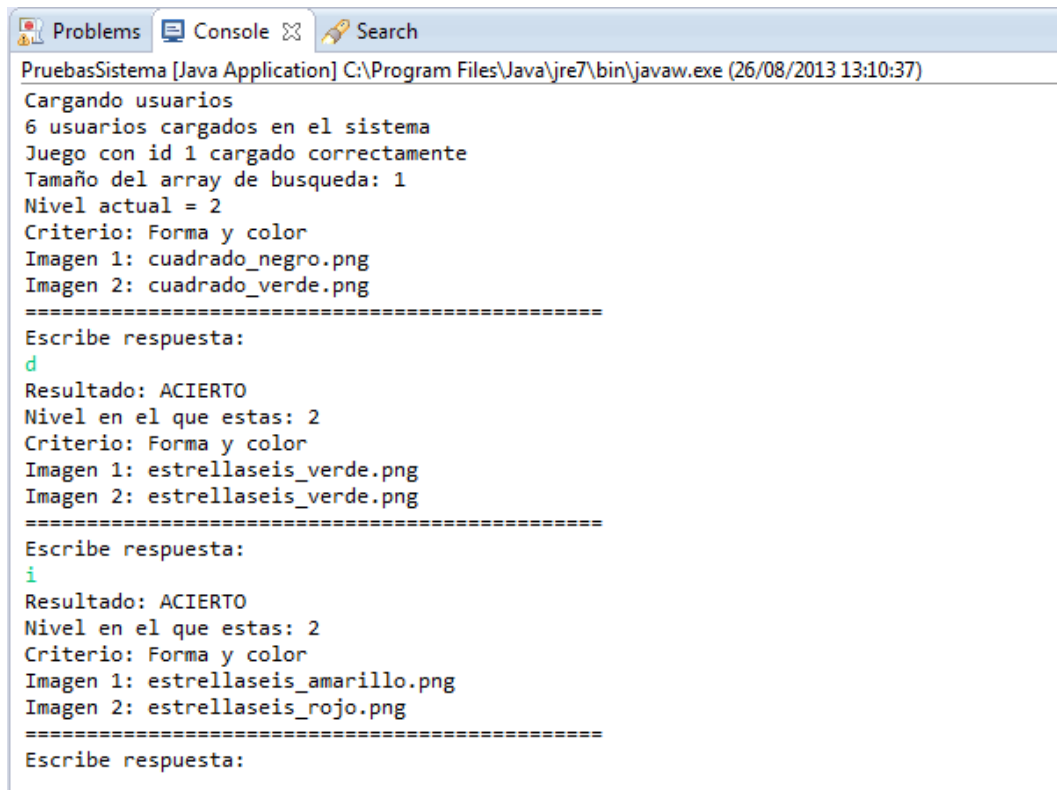
El resultado de estas pruebas fue satisfactorio, y además contribuyó a la mejora del desarrollo en cuanto a qué se encontraron errores a causa de no controlar errores y excepciones. Todos ellos se solucionaron y se repitieron las pruebas que los habían detectado para comprobar que se habían solucionado correctamente.

6.2.3 Pruebas del sistema

Este tipo de pruebas se han realizado en dos momentos clave del proyecto: al final del desarrollo de toda la lógica del programa y al final del desarrollo de la interfaz.

Para el primer caso se hizo un programa mediante el cual se puede interactuar con el juego por línea de comandos, con la salvedad de que en vez de mostrarse las imágenes se muestra el

nombre de éstas. Sin embargo esto no supone ningún problema porque, como ya se ha comentado anteriormente, el nombre de las imágenes es *forma_color* por lo que no es necesaria más información para saber si coinciden o no. En la figura 6.3 se muestran tres ejercicios.



```
Problems Console Search
PruebasSistema [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (26/08/2013 13:10:37)
Cargando usuarios
6 usuarios cargados en el sistema
Juego con id 1 cargado correctamente
Tamaño del array de busqueda: 1
Nivel actual = 2
Criterio: Forma y color
Imagen 1: cuadrado_negro.png
Imagen 2: cuadrado_verde.png
=====
Escribe respuesta:
d
Resultado: ACIERTO
Nivel en el que estas: 2
Criterio: Forma y color
Imagen 1: estrellaseis_verde.png
Imagen 2: estrellaseis_verde.png
=====
Escribe respuesta:
i
Resultado: ACIERTO
Nivel en el que estas: 2
Criterio: Forma y color
Imagen 1: estrellaseis_amarillo.png
Imagen 2: estrellaseis_rojo.png
=====
Escribe respuesta:
```

Figura 6.3: Pruebas del juego en línea de comandos

Como se puede observar en la figura anterior, se proporciona la misma información que en la pantalla correspondiente de la interfaz gráfica. A continuación se detallan los aspectos probados gracias a la ejecución del juego en línea de comandos.

- **Funcionamiento general del juego:** El aspecto fundamental de la prueba, que engloba todos los siguientes puntos. Se comprueba por ejemplo, que tras la respuesta, independientemente de su resultado se genera un nuevo ejercicio y se llevan a cabo las comprobaciones pertinentes de cambio de nivel o criterio.
- **Reglas de los niveles:** Se comprueba que se aplica correctamente toda la configuración de los niveles del juego leída en el archivo XML, como mostrar el criterio en pantalla o avisar del cambio del mismo al usuario.
- **Cambios de criterio:** Se comprueba que el cambio de criterio se realiza en el momento correcto y a un criterio distinto del que se tiene en el momento. Se comprueba también que el cambio de criterio provoca el cambio en la carga de imágenes.
- **Cambios de nivel:** Mediante diversas ejecuciones en las que se realizan un número determinado de aciertos o fallos para subir o bajar de nivel se observa que todos ellos se realizan de forma correcta. Dado que el salto entre el nivel 5 y 6 es manual y por tanto el sistema nunca lo realizará, se han hecho pruebas partiendo del nivel 2 y del nivel 6. Un error que se encontró es que el programa fallaba cuando intentaba subir de nivel más allá del 10,

ya que no hay más niveles, sin embargo no estaba controlado en la función que realiza los cambios de nivel. Gracias a esta prueba, este pequeño desliz fue corregido antes de que integrarlo con la interfaz gráfica.

- **Generación de resultados:** Finalmente se comprueba que los resultados que genera la aplicación al final de la ejecución del juego son los correspondientes a las respuestas que se han dado en los ejercicios y estos se graban correctamente en la base de datos.

Posteriormente a la unión de la lógica del juego con la interfaz se realizaron las mismas pruebas. Sin embargo la ventaja de haber realizado anteriormente la prueba en modo texto es que se pudo dejar a un lado la comprobación del funcionamiento para centrar la atención en los componentes de la pantalla, y el comportamiento de cada uno de ellos. Por ejemplo, se detectó se montaban el texto cuando se avisaba de cambio de nivel y el criterio era “Forma y color”, problema que se solucionó dando mayor distancia entre ambos elementos.

Otro de los problemas que se detectó al realizar las pruebas en la mesa con la interfaz gráfica fue que al cambiar de pantalla no se estaba haciendo de la forma adecuada y esto ralentizaba la aplicación, ya que las pantallas anteriores quedaban en un segundo plano pero se seguían destinando recursos a ellas. La solución a dicho problema fue añadir a los métodos que realizaban el cambio de pantalla el código necesario para eliminar las referencias a dichas pantallas y de esta forma centrar toda la atención en la actual.

6.2.4 Pruebas con usuarios

Las últimas pruebas realizadas a la aplicación fueron con usuarios reales sin ninguna familiarización previa con la aplicación. Se pretende comprobar, entre otros, los siguientes aspectos:

- **Rapidez de adaptación a la aplicación:** familiarización con cada una de las pantallas que las componen y manejo de cada una de ellas.
- **Comprensión del juego:** entender cómo se juega y para qué sirve la información que se muestra en la pantalla.

Las pruebas se realizaron con dos usuarios, un terapeuta y una niña de 9 años diagnosticada con TDAH, ambos sobre la mesa multicontacto, por lo que también es importante observar la adaptación de la persona con un dispositivo que no está acostumbrado a manejar. Se llevaron a cabo en dos días en los que se simulaban sesiones reales de terapia, el psicólogo con la niña para finalmente extraer las conclusiones que se citan a continuación.

- En cuanto al diseño de la interfaz gráfica, el psicólogo ha concluido que su sencillez contribuye a que sea el propio niño el que pueda manejar la aplicación, por ejemplo buscándose en el panel de usuarios gracias al nombre y a la foto.
- No se ha podido demostrar que se obtengan resultados de mejora de la atención, ya que se trata de un tratamiento a largo plazo, no abarcable en este proyecto. Sin embargo dado que la aplicación es una mejora de la ya comentada CAPTAIN, es esperable que el uso a largo plazo en una terapia real contribuya a la mejora de la atención por parte de los usuarios.

- La adaptación de la niña al dispositivo ha sido muy buena y ha despertado rápidamente su interés, por tratarse de algo nuevo. Esto es un aspecto muy positivo, ya que captar la atención del niño es fundamental.
- En muchos de los ejercicios se responde casi antes de poder asimilar las imágenes que se le presentan. Es importante recordar que una de las características del TDAH es la impulsividad, por lo que no es extraño. Se propone incorporar la siguiente mejora: Detección de respuestas muy rápidas e inclusión de un sistema que bloquee los botones cierto tiempo para forzar que el usuario centre su atención en las imágenes.
- Se detecta que la explicación del juego a la niña es algo costosa. Se propone incorporar un ejemplo guiado es decir, varios ejercicios en los que la aplicación vaya explicando al niño los elementos de la pantalla y lo que tiene que hacer, incluyendo voz y ejercicios sencillos que tiene que responder.
- Se propone también la inclusión de sonidos (distintos para acierto y error) de retroalimentación y refuerzo del paciente en los ejercicios.
- Se llevaron a cabo tres juegos de cinco minutos cada uno, con un pequeño descanso entre ellos, y se observó que en el tercer juego la niña se distraía con mayor facilidad. Al volver a jugar al día siguiente se observaron los mismos resultados: buena atención al principio y mayor distracción al final, de lo que el psicólogo concluyó que en cierto modo es normal, ya que los niños con TDAH tienden a distraerse realizando la misma acción durante mucho tiempo. Esto también ocurre con CAPTAIN, por lo que en la terapia real se van alternando juegos y ejercicios cada poco tiempo.

A modo de conclusión, comentar que las pruebas con los usuarios resultaron muy satisfactorias ya que se han extraído posibles mejoras que incorporar a la aplicación y se ha considerado útil para utilizar en una terapia real. En el siguiente capítulo se comentarán las conclusiones generales del proyecto que incluyen las extraídas en esta fase.

7

Conclusiones y trabajo futuro

Este último capítulo está dedicado a las conclusiones personales extraídas a lo largo de todo el proyecto, por lo que su redacción es en primera persona, al contrario que en el resto de la memoria.

En cuanto al cumplimiento de los objetivos y los requisitos, destacar que se han cumplido satisfactoriamente. El objetivo principal era crear una aplicación ejecutable sobre una mesa multicontacto que sirviera como complemento a la terapia con niños con TDAH ayudando a entrenar su atención y aligerando la labor del terapeuta.

He desarrollado un sistema capaz de dar soporte no sólo a la presentación de juegos y la variación automática de niveles, sino también a la visualización de resultados y el seguimiento de cada paciente por parte del terapeuta mediante los históricos. Por todo esto, puedo concluir que este objetivo se ha cumplido satisfactoriamente, como se vio en las pruebas de usuario.

Estoy muy satisfecha con el producto final obtenido, si bien es cierto que gracias a dichas pruebas hemos encontrado mejoras que lo harán más útil tanto para el terapeuta como para el niño, por lo que una vez concluido el trabajo tengo la intención de seguir ampliando y mejorando este proyecto.

Una de las mejoras que me parecen más importantes es la de la interfaz gráfica. En el anexo [A](#) se muestran los prototipos de la nueva interfaz diseñada con la ayuda de una persona especialista en ello ¹. Con esta nueva interfaz se dará un aspecto más profesional a la aplicación.

Además me gustaría incorporar las funcionalidades que se sugirió el terapeuta en las pruebas con los usuarios como el control de impulsos mediante el bloqueo de los botones y el ejemplo guiado. También mejorar algunos de los aspectos que aunque se han implementado, no funcionan tan bien como cabría esperar, como la limitación del tiempo de respuesta y ocultar las imágenes pasados ciertos segundos para dificultar el juego.

Como última mejora futura por el momento también me parece una buena idea la adapta-

¹Laura Andina, diseñadora en Tecnológica Soluciones Avanzadas SL

ción de la aplicación a otros sistemas operativos como iOS y Android, con la intención de que puedan ser utilizados en tabletas y *smartphones*. En Android únicamente habría que cambiar la interfaz dado que la lógica de estas aplicaciones se realiza en Java, sin embargo para iOS habría que implementar de nuevo prácticamente la aplicación entera, salvo la base de datos, que sería exactamente igual. Se cree que el uso de la aplicación a través de estos dispositivos puede motivar a niños y adolescentes a interactuar con los juegos incluso fuera de la terapia.

Atendiendo al desarrollo técnico del proyecto, debo reconocer que al principio me costó mucho sintetizar todas las ideas que tenía, empezar a darle forma y realizar el diseño final, pero en el momento que lo conseguí la implementación fue bastante rápida, con lo que destaco la importancia de las fase de análisis y diseño como parte fundamental de todo proyecto.

No solo he aprendido a desarrollar un proyecto desde el principio hasta el final sino que también he adquirido nuevos conocimientos respecto a tecnologías y bibliotecas útiles para la interacción con dispositivos multicontacto.

Finalmente, dar las gracias de nuevo a las personas que han contribuido al éxito del proyecto y a los usuarios que han realizado las pruebas, ya que su ayuda ha sido fundamental. Me ha gustado mucho hacer este proyecto ya que siento que puede ser de gran ayuda para muchas personas y porque he trabajado con especialistas en otros campos, por lo que también he aprendido un poquito de sus materias.

Como ya he comentado a lo largo del proyecto, la colaboración de especialistas en distintas disciplinas me parece fundamental para el desarrollo de este tipo de proyectos, ya que si se consiguen juntar los conocimientos de varias personas, la fiabilidad del producto construido seguro que es mayor que trabajando por separado. Creo que este proyecto es un ejemplo a pequeña escala, ya que he incluido conceptos teóricos de psicología y tanto el terapeuta como yo nos hemos beneficiado de los conocimientos puestos en común.

Bibliografía

- [1] E. Cardo and M. ServeraBarceló. Prevalencia del trastorno de déficit de atención e hiperactividad. *Rev. Neurol*, 40 (Supl1): S11-5, 2005.
- [2] 5th ed. Dsm-v. *Asociación Estadounidense de Psiquiatría*, 2013.
- [3] Psicoactiva - <http://www.psicoactiva.com/software.htm>.
- [4] Brain arena - <http://www.brainarena.com/countryselect?countrycode=es>.
- [5] Presentación de la plataforma neuronup - <http://www.abc.es/20120429/ciencia/abci-verdadero-brain-training-201204271632.html>.
- [6] Plataforma mt4j - <http://www.mt4j.org/>.
- [7] Herramientas proporcionadas por windows para programación en dispositivos multicontacto - <http://www.microsoft.com/en-us/pixelsense/softwareplatform.aspx>.
- [8] Joyce Karreman, Thea Geest van der, and Esmee Buursink. Accessible website content guidelines for users with intellectual disabilities. *Journal of Applied Research in Intellectual Disabilities*, 20(6):510–518, 2007.
- [9] Plataforma jdom - <http://www.jdom.org/downloads/>.
- [10] Mt4j, preguntas más frecuentes - <http://www.mt4j.org/mediawiki/index.php/faq>.



Prototipo de interfaz mejorada

Como trabajo futuro se ha propuesto la mejora de la interfaz, para lo que se ha pedido orientación a una persona especialista en el diseño de interfaces gráficas ¹.

Esta persona ha propuesto algunos cambios sobre la interfaz inicial siempre teniendo en cuenta los principios de diseño de interfaces para usuarios especiales que ya se comentaron. Básicamente con la nueva interfaz se utiliza una combinación de colores mejor elegidos que los originales y un tipo de letra diferente, haciendo la interfaz más agradable a la vista y con aspecto más profesional.

A continuación se incluyen los prototipos de la nueva interfaz. En las figuras [A.1](#), [A.2](#) y [A.3](#) se pueden ver ejemplos de la pantalla del juego, mientras que en la figura [A.4](#) se muestra la paleta de colores a utilizar tanto en estas pantallas como en el resto de la aplicación.

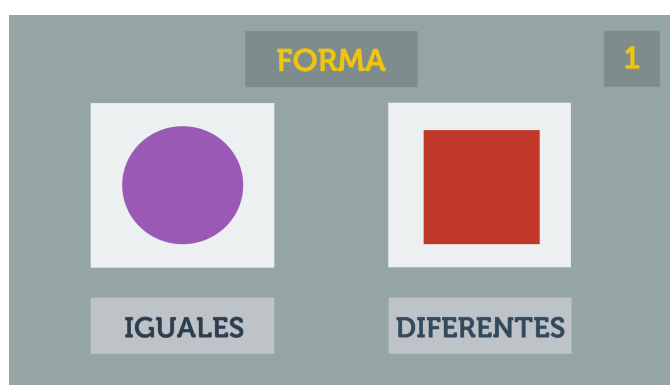


Figura A.1: Prototipo de la interfaz mejorada

¹Laura Andina, diseñadora en Tecnológica Soluciones Avanzadas SL

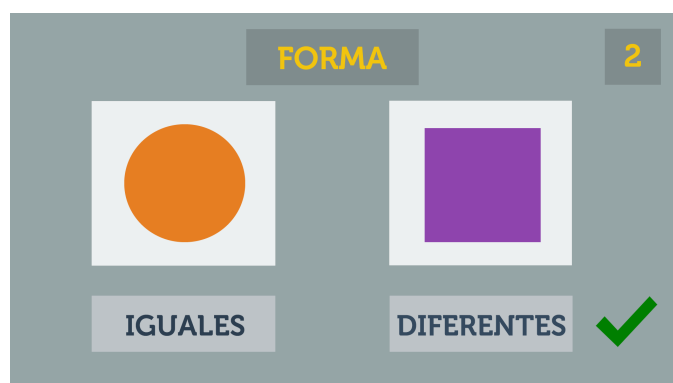


Figura A.2: Prototipo de la interfaz mejorada

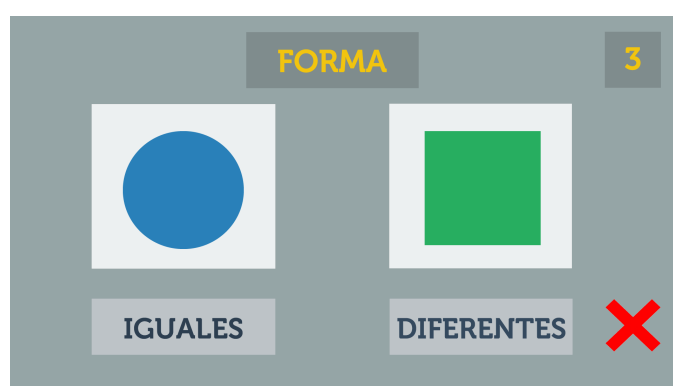


Figura A.3: Prototipo de la interfaz mejorada



Figura A.4: Nueva combinación de colores y tipografía

B

Tablas de atributos de las clases

En este anexo se incluyen a modo de documentación las tablas de atributos de las clases a las que se hace referencia en el capítulo 4.

Cuadro B.1: Atributos de la clase Sistema

Atributo	Descripción
mapUsuarios	Almacena los usuarios del sistema cargados del fichero XML “listadoUsuarios.xml” que se carga al crear el objeto.
mapJuegos	Almacena los juegos que ofrece el sistema, cargados del archivo “listadoJuegos.xml”.
ficherosJuegos	Relación de los identificadores de los juegos con los archivos que contienen su configuración.
objListUsuarios	Objeto de la clase ListadoUsuariosXML.
objListJuegos	Objeto de la clase ListadoJuegosXML.
objUsuariosXML	Objeto de la clase UsuariosXML.
objSesXML	Objeto de la clase SesionXML.
objResXML	Objeto de la clase ResultadosXML.
objUtiles	Objeto de la clase Utiles.

Cuadro B.2: Atributos de la clase Usuario

Atributo	Descripción
idUsuario	Guarda la id del usuario que se genera al crear el objeto.
nombre	Nombre del usuario.
apellido1	Primer apellido del usuario.
apellido2	Segundo apellido del usuario.
rutaFoto	Ruta de la imagen asociada al usuario a partir de la carpeta del proyecto.
fNac	Fecha de nacimiento.
diagnostico	Diagnóstico del paciente, proporcionado por el terapeuta.
cInicio	Fecha de inicio de la terapia.
observaciones	Posibles observaciones que quiera añadir el terapeuta.
infoSesiones	Listado de sesiones llevadas a cabo con el usuario.
historial	Historial de las sesiones del usuario.

Cuadro B.3: Atributos de la clase Historial

Atributo	Descripción
orden	Orden cronológico de la sesión.
idSesion	id de la sesión a la que hace referencia.
fecha	Fecha de la sesión.
tiempoMedioResp	Tiempo medio de respuesta de la sesion.
porcentajeAciertos	Porcentaje de aciertos en la sesión.

Cuadro B.4: Atributos de la clase InfoResultado

Atributo	Descripción
idJuego	Identificador del juego que se ha llevado a cabo.
cInicio	Fecha y hora de inicio.
cFin	Fecha y hora de fin.
nivelInicio	Nivel de inicio.
nivelFin	Nivel alcanzado al final del juego.
totalResp	Total de ejercicios respondidos durante el desarrollo del juego.
nAciertos	Número de aciertos alcanzados en el juego.
nFallosCom	Número de fallos por comisión.
nFallosOm	Número de fallos por omisión.
tiempos	Lista en la que, a lo largo del desarrollo del juego se van almacenando los valores de las respuestas y los tiempos.

Cuadro B.5: Atributos de la clase RegistroTiempo

Atributo	Descripción
rsRes	Respuesta dada por el usuario. Puede ser acierto, fallo por comisión o fallo por omisión.
tiempo	Tiempo que ha tardado en darse la respuesta.

Cuadro B.6: Atributos de la clase Sesion

Atributo	Descripción
idSesion	Identificador de la sesion.
idUsuario	Identificador del usuario al que pertenece la sesión.
cInicio	Fecha y hora de inicio.
duracion	Duración en segundos de la sesión.
extracto	Información sintetizada de los juegos llevados a cabo en la sesión y su nivel de inicio y fin.
comentarios	Posibles comentarios a añadir por el terapeuta.

Cuadro B.7: Atributos de la clase Juego

Atributo	Descripción
idJuego	Identificador del juego.
rutaDatos	Ruta en la que se encuentran todos los datos necesarios para el juego, en este caso las imágenes.
nombreJuego	Nombre del juego.
niveles	Lista con la configuración de los niveles del juego.
imagenes	Lista con las imágenes que maneja el juego.

Cuadro B.8: Atributos de la clase Nivel

Atributo	Descripción
numNivel	Numero de nivel.
avisoCC	Indica si se avisa al usuario al cambiar de criterio.
mostrarC	Indica si se muestra o no el criterio en pantalla.
segOcultarImg	Indica los segundos a partir de los cuales se ocultan las imágenes.
segLimRes	Indican los segundos límite para dar la respuesta.

Cuadro B.9: Atributos de la clase MotorJuego

Atributo	Descripción
juego	Configuración del juego sobre el que va a actuar el motor.
usuario	Usuario que lleva a cabo el juego.
resultado	Objeto InfoResultado generado con los resultados del juego.
orden	Orden secuencial del ejercicio.
nivelActual	Nivel en el que se encuentra el usuario. Se actualiza en cuando corresponde de la aplicación de las reglas.
nAciertosSeguidos	Número de aciertos seguidos. Se pone a cero si llega un fallo.
noErrorOmSeg	Número de ejercicios seguidos en los que no ha habido fallo de omisión.
nErroresOm	Número de errores de omisión.
nAciertosCambioCri	Contador de aciertos. Cuando llega a 8 se cambia el criterio y se pone a cero.
imagen1	Imagen 1 seleccionada aleatoriamente.
imagen2	Imagen 2 seleccionada según las reglas explicadas.
critForma	Indica si la forma infuye en el criterio.
critColor	Indica si el color infuye en el criterio.

